Journal of Computational Physics **162,** 301–337 (2000)
doi:10.1006/jcph.2000.6537, available online at http://www.idealibrary.com on IDEAL®

A Coupled Level Set and Volume-of-Fluid Method for Computing 3D and Axisymmetric Incompressible Two-Phase Flows

Mark Sussman[1] and Elbridge Gerry Puckett[2]

Department of Mathematics, Florida State University, Tallahassee, Florida 32306; Department of Mathematics, University of California, Davis, Davis, California 95616
E-mail: sussman@math.fsu.edu

Received January 15, 1999; revised March 28, 2000

We present a coupled level set/volume-of-fluid (CLSVOF) method for computing 3D and axisymmetric incompressible two-phase flows. This method combines some of the advantages of the volume-of-fluid method with the level set method to obtain a method which is generally superior to either method alone. We present direct comparisons between computations made with the CLSVOF method and computations made with the level set method, the volume-of-fluid method, and the boundary integral method. We also compare our computations to the exact solution for an oscillating ellipse due to Lamb and experimental results obtained for a rising gas bubble in liquid obtained by Hnat and Buckmaster. Our computational examples focus on flows in which surface tension forces and changes in topology are dominant features of the flow. © 2000 Academic Press

Key Words: level set; volume-of-fluid; curvature; incompressible.

1. INTRODUCTION

In this work we present a coupled level set and volume-of-fluid method (CLSVOF) for computing 3D and axisymmetric incompressible two-phase flows, with and without viscosity. Incompressible two-phase flow is often difficult to model computationally, since the density ratio at the free-surface between the gas and liquid can be 1000 : 1 (e.g., air–water) or more. Furthermore, complications may arise when surface tension is present and/or when a change of topology occurs. An example of the flows we compute is shown

[1] Work supported in part by the National Science Foundation under Contract DMS 97-06847.

[2] Work supported in part by the National Science Foundation under Contract DMS 97-06847 and by the Mathematical, Information and Computational Science Division of the U.S. Department of Energy's office of energy research #DE-FG03-95ER25271 under Contract.

0021-9991/00 $35.00
Copyright © 2000 by Academic Press
All rights of reproduction in any form reserved.

in Fig. 15 in Section 6 where we show the results of a 4-mm air bubble rising to the surface of an air/water interface and then bursting due to stiff surface tension effects. The jet then breaks up due to capillary instabilities, emitting satellite drops.

Our development of the CLSVOF method has been partially motivated by our desire to model microscale jetting devices, such as ink-jet print heads. In a typical microscale jetting device, a liquid is ejected from a nozzle which has a characteristic length of 1–50 microns. The liquid typically forms into a lead drop which is roughly spherical, followed by a long, thin cylindrical tail. Due to the length scales in the problem, surface tension is the dominant force affecting the dynamics of this lead drop/tail system. In particular, the tail usually separates from the lead drop and undergoes a Rayleigh capillary instability, breaking into two or more smaller "satellite drops;" e.g., see Fig. 4 in Section 4. Therefore, it is of critical importance to model surface tension accurately, if one wishes to correctly model this problem.

Throughout this paper, we will primarily compare the CLSVOF method to either the level set (LS) method as described in [29] or the volume-of-fluid (VOF) method as described in [22]. We also make a comparison with results computed with the boundary integral method described in [32]. We note, that although we have focused on these three methods, there are a variety of other numerical models proposed for solving incompressible free-surface problems such as boundary integral methods [8, 9, 20], front tracking methods [35], and particle-in-cell methods [25].

In the LS method [17, 21, 29, 32, 33], a smooth function $\phi(r, z, t)$—called the *level set function*—is used to represent the free surface. Liquid regions are regions in which $\phi(r, z, t) > 0$ while gas regions are regions in which $\phi(r, z, t) < 0$. The free-surface is implicitly represented by the set of points in which $\phi(r, z, t) = 0$. One of the advantages of the LS method is its simplicity, especially when computing the curvature κ of the interface. Typically the level set function $\phi(r, z, t)$ is maintained as the signed distance to the free surface; i.e., $\phi(r, z, t) = -d$ in the gas and $\phi(r, z, t) = +d$ in the liquid where $d = d(t)$ is the shortest distance from the point (r, z) to the free-surface at time t; e.g., see Sussman *et al.* [33]. From such a representation of the free-surface, the unit normal vector $\mathbf{n}$ normal and mean curvature κ are simply

$$\mathbf{n} = \frac{\nabla\phi}{|\nabla\phi|}, \tag{1}$$

and

$$\kappa = \nabla \cdot \frac{\nabla\phi}{|\nabla\phi|}, \tag{2}$$

respectively. In particular, note that when $|\nabla\phi| = 1$, the discretization for κ in 2 reduces to a discrete Laplacian of ϕ. On the other hand, the LS method has the disadvantage that the discretization of the equation to advect the level set function,

$$\phi_t + u\phi_r + v\phi_z = 0, \tag{3}$$

where (u, v) is the underlying velocity field, is prone to more numerical error than front-tracking methods or VOF methods when the interface experiences severe stretching or tearing. A common symptom is loss/gain of mass. We remark that in an incompressible flow $\nabla \cdot \boldsymbol{U} = 0$ and hence (3) is equivalent to the conservation law

$$\phi_t + (u\phi)_r + (v\phi)_z = 0. \tag{4}$$

However, although ϕ can be discretely conserved via (4), the mass enclosed by the zero level set of ϕ is not conserved. Other related problems, reported by Rider and Kothe [24], can occur if the interface undergoes severe stretching and tearing.

One motivation for developing the CLSVOF method is this loss of mass problem; for example, computing a three dimensional ink-jet problem, we are limited to relatively coarse computational grids. On a coarse grid, a LS method will tend to lose so much mass that the tail disappears before it breaks up into satellite drops. Prior to developing the CLSVOF method, we tried computing with the velocity field on a coarse grid and the level set function on a fine grid. Unfortunately, this is an inefficient idea since the redistance step would have to be done on the fine grid and the surface tension time step constraint would have to be derived relative to the mesh size on the fine grid.

Besides the LS approach, we have considered the volume-of-fluid method (VOF). In the volume-of-fluid method (VOF) [1, 11, 23], the volume fraction $F(\Omega, t)$ is used to represent the free surface. Typically, Ω represents a computational cell Ω_{ij}; e.g., $\Omega_{ij} = \{(r, z) \mid r_i \le r \le r_{i+1}$ and $z_j \le z \le z_{j+1}\}$. If $F(\Omega, t) = 1$, then the region Ω is all liquid. If $F(\Omega, t) = 0$, then the region Ω is all gas. If $0 < F(\Omega, t) < 1$, then Ω contains both gas and liquid. One can define the volume fraction function $F(\Omega, t)$ in terms of the level set function $\phi(r, z, t)$. Since we have $\phi > 0$ in the liquid and $\phi < 0$ in the gas, one can define $F(\Omega, t)$ as

$$F(\Omega, t) = \frac{1}{|\Omega|} \int_{\Omega} H(\phi(r, z, t)) r \, dr \, dz, \tag{5}$$

where H is the Heaviside function,

$$H(\phi) = \begin{cases} 1 & \text{if } \phi > 0 \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

An advantage of representing the free surface as volume fractions is the fact that one can write accurate algorithms for advecting the volume fraction function so that mass is conserved while still maintaining a sharp representation of the interface. For example, Pilliod and Puckett [22] developed second order volume-of-fluid advection methods which accurately compute the rotation of a notched disk while maintaining mass conservation. However, a disadvantage of the VOF method is the fact that it is difficult to compute accurate local curvatures from volume fractions. This is because the volume fractions transition sharply in regions of the free surface. Standard VOF methods compute the curvature by first mollifying the volume fractions in a special way (see [1, 11]). We have experienced difficulty with this approach. If one does not smooth enough, then the curvature for even a circle will be highly oscillatory. This would spell disaster for the "stationary bubble test."

If one smooths too much then the numerical algorithm will not "see" changes in curvature along the free surface (since too much smoothing has the effect of making the curvature constant along the free surface). This can spell disaster for dynamic problems such as the zero gravity drop oscillation problem (see Subsection 6.2). In the CLSVOF method, we do not smooth the curvature at all; instead the curvature is obtained via finite differences of the level set function which in turn is derived from the level set function and volume-of-fluid function at the previous time step.

In this work, we couple the LS method with the volume-of-fluid method; early work along these lines was done by Bourlioux [10]. However, our approach to coupling the LS method with the volume-of-fluid method differs from Bourlioux's. In addition, we also combine a

coupled level set/volume-of-fluid advection method with the equations for incompressible two-phase flow.

We show that we have comparable mass conservation properties as with other second order volume-of-fluid advection methods [22] and we also show that we can accurately compute surface tension driven flows by coupling the LS method with the volume-of-fluid method.

2. GOVERNING EQUATIONS

The equations of motion for incompressible two-phase flow can be written as

$$\boldsymbol{U}_t + \boldsymbol{U} \cdot \nabla \boldsymbol{U} = -\frac{\nabla p}{\rho(\phi)} + \frac{1}{\rho(\phi)} \nabla \cdot (2\mu(\phi) D) - \frac{1}{\rho(\phi)} \gamma \kappa(\phi) \nabla H(\phi) + \boldsymbol{F} \tag{7}$$

$$\nabla \cdot \boldsymbol{U} = 0, \tag{8}$$

and

$$\phi_t + \boldsymbol{U} \cdot \nabla \phi = 0. \tag{9}$$

ϕ is the level set function which is positive in the liquid and negative in the gas. The governing equation for the level set function (9) states that ϕ remains constant on particle paths; i.e., if the zero level set is initialized as the free-surface between the liquid and gas, then the zero level set will always represent the free-surface. From the level set function, one can derive the density $\rho(\phi)$, viscosity $\mu(\phi)$, and curvature $\kappa(\phi)$. Density and viscosity are written as

$$\rho(\phi) = \rho_g(1 - H(\phi)) + \rho_l H(\phi) \tag{10}$$

$$\mu(\phi) = \mu_g(1 - H(\phi)) + \mu_l H(\phi), \tag{11}$$

where $H(\phi)$ is the Heaviside function defined by (6). The local mean curvature can be written as

$$\kappa(\phi) = \nabla \cdot \frac{\nabla \phi}{|\nabla \phi|}. \tag{12}$$

D is defined as the rate of deformation tensor,

$$D = (\nabla \boldsymbol{U}) + (\nabla \boldsymbol{U})^T. \tag{13}$$

$\boldsymbol{F}$ is a body force; in our implementations, $\boldsymbol{F}$ represents the force due to gravity $\boldsymbol{F} = (0, g)$. ρ_g, ρ_l, μ_g, μ_l, and γ are defined to be the gas density, liquid density, gas viscosity, liquid viscosity, and surface tension coefficient, respectively.

The fact that the surface tension force,

$$\frac{1}{\rho(\phi)} \gamma \kappa(\phi) \nabla H(\phi), \tag{14}$$

can be cast as a body force is due to the work of Brackbill *et al.* [11] and more recently, for the LS method, due to the work of [12].

We remark that when we discretize the level set equation (9), we shall simultaneously solve the following equation for the volume-of-fluid function F,

$$F_t + \nabla \cdot (\boldsymbol{U}F) = 0. \tag{15}$$

At $t = 0$, F will be initialized in each computational cell Ω_{ij},

$$\Omega_{ij} = (r, z) \mid r_i \le r \le r_{i+1} \qquad \text{and} \qquad z_j \le z \le z_{j+1}, \tag{16}$$

to be

$$F_{ij} = \frac{1}{\Delta r \, \Delta z} \int_{\Omega_{ij}} H(\phi(r, z, 0)) r \, dr \, dz. \tag{17}$$

Here, Δr and Δz are defined as $r_{i+1} - r_i$ and $z_{i+1} - z_i$, respectively.

2.1. Projection methodology. The method used to solve for velocity and pressure is a variable density approximate projection method described by [4, 23]. We rewrite (7) as

$$\boldsymbol{U}_t + \frac{1}{\rho(\phi)} \nabla p = \boldsymbol{V}(\boldsymbol{U}, \phi). \tag{18}$$

We then take the divergence of both sides of (18) and use the fact that $\nabla \cdot \boldsymbol{U}_t = 0$ in order to reduce (7) and (8) into a single equation for pressure,

$$\nabla \cdot \frac{1}{\rho} \nabla p = \nabla \cdot \boldsymbol{V}. \tag{19}$$

After solving (19) for ∇p the updated value for $\boldsymbol{U}_t$ is

$$\boldsymbol{U}_t = \boldsymbol{V} - \nabla p / \rho. \tag{20}$$

For future reference, we define the projection operator $\boldsymbol{P}$ as

$$\boldsymbol{U}_t \equiv \boldsymbol{P}(\boldsymbol{V}). \tag{21}$$

Combining (21) and (20) yields

$$\nabla p / \rho = \boldsymbol{V} - \boldsymbol{U}_t \equiv \boldsymbol{V} - \boldsymbol{P}(\boldsymbol{V}) \equiv (I - P)(\boldsymbol{V}). \tag{22}$$

3. THE CLSVOF ADVECTION ALGORITHM

In this section, we describe how to advance the free surface using the coupled level set volume-of-fluid (CLSVOF) advection algorithm. We shall describe the details for the axisymmetric case. The 3D algorithm follows analogously. We shall discretize our variables on a uniform grid with grid spacing of $\Delta r = \Delta z$. The discrete level set function $\phi^n_{i,j}$ and discrete volume fraction $F^n_{i,j}$ are located at cell centers. The motion of the free surface is determined by the velocity field derived from the equations for incompressible two-phase

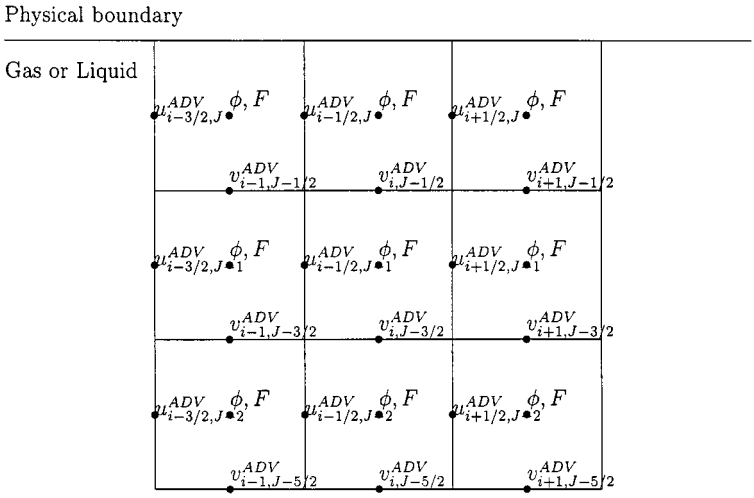

FIG. 1. Diagram of where the discretely divergence-free edge velocity field $\boldsymbol{U}^{ADV}$, level set function ϕ, and volume-of-fluid function F are located in relation to the computational grid and the physical boundary.

flow. The discrete velocity field is defined at cell edges $u_{i+1/2,\,j}$ and $v_{i,\,j+1/2}$, and satisfies the discrete divergence free condition,

$$D^{MAC}\boldsymbol{U} = \frac{(ru)_{i+\frac{1}{2},\,j} - (ru)_{i-\frac{1}{2},\,j}}{r_i\,\Delta r} + \frac{v_{i,\,j+\frac{1}{2}} - v_{i,\,j-\frac{1}{2}}}{\Delta z}. \tag{23}$$

A diagram of where the discrete variables are located in relation to the computational grid is shown in Fig. 1. J represents the index of the computational cell closest to the top physical boundary.

The equations governing the interface motion are

$$\phi_t + \nabla \cdot (\boldsymbol{U}\phi) = 0 \tag{24}$$

and

$$F_t + \nabla \cdot (\boldsymbol{U}F) = 0. \tag{25}$$

Remark. The conservative formulation of the level set equation (24) is equivalent to (3) since $\nabla \cdot \boldsymbol{U} = 0$.

We shall assume that the level set function $\phi^0_{i,\,j}$ is initialized as the signed normal distance from the initial position of the free surface. The volume fraction function $F^0_{i,\,j}$ shall be initialized as the fraction of liquid fluid contained in cell $(i,\,j)$. In other words,

$$F^0_{i,\,j} = \frac{1}{\Delta r\,\Delta z}\int_{\Omega_{ij}} H(\phi(r,z,0))r\,dr\,dz, \tag{26}$$

where

$$\Omega_{ij} = (r,z) \mid r_i \le r \le r_{i+1} \qquad \text{and} \qquad z_j \le z \le z_{j+1}. \tag{27}$$

Given $\phi^n_{i,j}$, $F^n_{i,j}$, and $\boldsymbol{U}$, we use a "coupled" second order conservative operator split advection scheme in order to find $\phi^{n+1}_{i,j}$ and $F^{n+1}_{i,j}$. For axisymmetric flows, the operator split algorithm for a general scalar s follows as

$$\tilde{s}_{i,j} = \frac{s^n_{i,j} + (\Delta t/r_i \Delta r)\big(r_{i-1/2} G_{i-1/2,j} - r_{i+1/2} G_{i+1/2,j}\big)}{1 - (\Delta t/r_i \Delta r)\big(r_{i+1/2} u_{i+1/2,j} - r_{i-1/2} u_{i-1/2,j}\big)}, \tag{28}$$

$$s^{n+1}_{i,j} = \tilde{s}_{i,j} + \frac{\Delta t}{\Delta z}\big(\tilde{G}_{i,j-1/2} - \tilde{G}_{i,j+1/2}\big) + \tilde{s}_{i,j}\big(v_{i,j+1/2} - v_{i,j-1/2}\big), \tag{29}$$

where $G_{i+1/2,j} = s_{i+1/2,j} u_{i+1/2,j}$ denotes the flux of s across the right edge of the (i, j)th cell and $\tilde{G}_{i,j+1/2} = \tilde{s}_{i,j+1/2} v_{i,j+1/2}$ denotes the flux across the top edge of the (i, j)th cell. For 3D flows, the operator split algorithm for a general scalar s follows as

$$\tilde{s}_{i,j,k} = \frac{s^n_{i,j,k} + (\Delta t/\Delta x)\big(G_{i-1/2,j,k} - G_{i+1/2,j,k}\big)}{1 - (\Delta t/\Delta x)\big(u_{i+1/2,j,k} - u_{i-1/2,j,k}\big)}, \tag{30}$$

$$\hat{s}_{i,j,k} = \frac{\tilde{s}_{i,j,k} + (\Delta t/\Delta y)\big(\tilde{G}_{i,j-1/2,k} - \tilde{G}_{i,j+1/2,k}\big)}{1 - (\Delta t/\Delta y)\big(v_{i,j+1/2,k} - v_{i,j-1/2,k}\big)}, \tag{31}$$

$$\bar{s}_{i,j,k} = \frac{\hat{s}_{i,j,k} + (\Delta t/\Delta z)\big(\hat{G}_{i,j,k-1/2} - \hat{G}_{i,j,k+1/2}\big)}{1 - (\Delta t/\Delta z)\big(w_{i,j,k+1/2} - w_{i,j,k-1/2}\big)}, \tag{32}$$

$$s^{n+1}_{i,j,k} = \bar{s} - \Delta t\left(\frac{\tilde{s}_{i,j,k}}{\Delta x}\big(u_{i+1/2,j,k} - u_{i-1/2,j,k}\big) + \frac{\hat{s}}{\Delta y}\big(v_{i,j+1/2,k} - v_{i,j-1/2,k}\big) + \frac{\bar{s}}{\Delta z}\big(w_{i,j,k+1/2} - w_{i,j,k-1/2}\big)\right). \tag{33}$$

Remarks.

- Although (28) and (29) are not in conservation form, the scalar s is still conserved since $u_{i+1/2,j}$ and $v_{i,j+1/2}$ satisfy (23). The form that we use to difference the fluxes in (28) and (29) was used by [23]. The 3D analogue represented by (30) thru (33) reduces to the two dimensional case when any one of the discrete derivatives u_x, v_y, or w_z is zero.
- The operator split procedure described above is made second order accurate by alternating the starting sweep direction at each time step, i.e., by employing "Strang splitting" [28].

The scalar flux $s_{i+1/2,j}$ is computed differently depending on whether s represents the level set function ϕ or the volume fraction function F. For the case when s represents the level set function ϕ we have the following representation for $s_{i+1/2,j}$ when $u_{i+1/2,j} > 0$,

$$s_{i+1/2,j} = s^n_{i,j} + \frac{\Delta r}{2}\left(1 - u_{i+1/2,j}\frac{\Delta t}{\Delta r}\right)\frac{s^n_{i+1,j} - s^n_{i-1,j}}{\Delta r} \tag{34}$$

and when $u_{i+1/2,j} < 0$,

$$s_{i+1/2,j} = s^n_{i+1,j} - \frac{\Delta r}{2}\left(1 + u_{i+1/2,j}\frac{\Delta t}{\Delta r}\right)\frac{s^n_{i+2,j} - s^n_{i,j}}{\Delta r}.$$

The above discretization is motivated by the predictor corrector method described in [5] and the references therein. The scalar flux $s_{i+1/2,j}$ is obtained by extrapolating s in both

space and time. Below, we show an example for the case when $u_{i+1/2,j} > 0$,

$$s_{i+1/2,j} \approx s_{i,j} + \frac{\Delta r}{2} s_{r,ij} + \frac{\Delta t}{2} s_{t,ij}. \tag{35}$$

For an operator split algorithm we only solve for one direction at a time. This means, for example, that we are solving

$$s_t + u s_r = 0.$$

We can substitute $s_{t,ij} = -u s_{r,ij}$ into (35) in order to obtain

$$s_{i+1/2,j} \approx s_{i,j} + \frac{\Delta r}{2}\left(1 - u\frac{\Delta t}{\Delta r}\right) s_{r,ij}.$$

If we replace u with $u_{i+1/2,j}$ and $s_{r,ij}$ with $(s^n_{i+1,j} - s^n_{i-1,j})/\Delta r$, then we recover (34).

For the case when s represents the volume-of-fluid function F we have the following representation for $s_{i+1/2,j}$ when $u_{i+1/2,j} > 0$,

$$s_{i+1/2,j} = \frac{\left(\int_{z_{j-1/2}}^{z_{j+1/2}} \int_{r_{i+1/2}-u_{i+1/2,j}\Delta t}^{r_{i+1/2}} H\left(\phi^{n,R}_{i,j}(r,z)\right) r\, dr\, dz\right)}{\left(r_{i+1/2} - (1/2)u_{i+1/2,j}\Delta t\right) u_{i+1/2,j}\Delta t \Delta z} \tag{36}$$

and when $u_{i+1/2,j} < 0$,

$$\frac{\left(\int_{z_{i-1/2}}^{z_{i+1/2}} \int_{r_{i+1/2}}^{r_{i+1/2}-u_{i+1/2,j}\Delta t} H\left(\phi^{n,R}_{i+1,j}(r,z)\right) r\, dr\, dz\right)}{\left(r_{i+1/2} - (1/2)u_{i+1/2,j}\Delta t\right)\left|u_{i+1/2,j}\right| \Delta t \Delta z}. \tag{37}$$

The term $\phi^{n,R}_{i,j}(r,z)$ found in (36) and (37) represents the linear reconstruction of the interface in cell (i, j). In other words, $\phi^{n,R}_{i,j}(r,z)$ has the form

$$\phi^{n,R}_{i,j}(r,z) = a_{i,j}(r - r_i) + b_{i,j}(z - z_j) + c_{i,j}. \tag{38}$$

The coefficients $a_{i,j}$, $b_{i,j}$, and $c_{i,j}$ are first chosen so that (38) represents the best fit line for the piece of the zero level set passing thru cell (i, j). In other words, a, b, and c minimize the error

$$E_{i,j} = \int_{r_{i-1/2}}^{r_{i+1/2}} \int_{z_{i-1/2}}^{z_{i+1/2}} H'(\phi)(\phi - a_{i,j}(r - r_i) - b_{i,j}(z - z_j) - c_{i,j})^2. \tag{39}$$

In order to solve for a, b, and c, we minimize the discretized error,

$$E^{\Delta r}_{i,j} = \sum_{i'=i-1}^{i+1} \sum_{j'=j-1}^{j+1} w_{i'-i,j'-j} H'_\epsilon(\phi_{i',j'})(\phi_{i',j'} - a_{i,j}(r_{i'} - r_i) - b_{i,j}(z_{j'} - z_j) - c_{i,j})^2. \tag{40}$$

The discrete weights $w_{r,s}$ are chosen so that (40) is an approximation to (39). For the computations we show, we have $w_{r,s} = 16$ for $r = s = 0$ and $w_{r,s} = 1$ for $r \neq 0$ or $s \neq 0$. We have tried other values for $w_{r,s}$ with little effect on the accuracy of the computation. $H'_\epsilon(\phi)$ represents the smoothed delta function with thickness ϵ; in our computations, we always

have $\epsilon = \sqrt{2}\Delta r$. The resulting equations for a, b, c as a result of minimizing (40) are a 3×3 linear system.

The intercept $c_{i,j}$ is corrected so that the line represented by (38) cuts out the same volume in cell (i, j) as specified by $F^n_{i,j}$. In other words, the following equation is solved for $c_{i,j}$,

$$\int_{z_{i-1/2}}^{z_{i+1/2}} \int_{r_{i-1/2}}^{r_{i+1/2}} H(a_{i,j}(r - r_i) + b_{i,j}(z - z_j) + c_{i,j}) r \, dr \, dz = F^n_{i,j}. \tag{41}$$

Since H is a Heaviside function defined as $H(\phi) = 1$ if $\phi > 0$ and $H(\phi) = 0$ otherwise, we solve (41) by use of the Newton iteration method. We remark that the algorithm is simplified by first rotating the grid axis so that the normal represented by $a_{i,j}$ and $b_{i,j}$ points away from the lower left hand corner of the (i, j) computational cell. The coefficients $a_{i,j}$, $b_{i,j}$, and $c_{i,j}$ are also rescaled so that $a^2_{i,j} + b^2_{i,j} = 1$ and the new intercept represents the normal distance to the lower left hand corner of the computational cell.

The integrals in (36) and (37) are evaluated by finding the volume cut out of the region of integration by the line represented by (38).

Remarks.

- In comparison to setting

$$a_{i,j} = \frac{1}{2\Delta r}(\phi_{i+1,j} - \phi_{i-1,j}) \tag{42}$$

$$b_{i,j} = \frac{1}{2\Delta z}(\phi_{i,j+1} - \phi_{i,j+1}), \tag{43}$$

the minimization procedure described above proved to be more accurate for simple tests such as the translation of a circle or Zalesak's problem.

- In our numerical procedure, we disallow partial volume fractions in cells in which $|\phi| > \Delta r$. This guarantees that we only reconstruct the interface in cells in which $|\phi| \le \Delta r$; thus the linear system that results from minimizing (40) will always have a solution.

The scalar flux $\tilde{s}_{i,j+1/2}$ is computed in the same manner as $s_{i+1/2,j}$. For the case when s represents the level set function ϕ, we have the following representation for $\tilde{s}_{i,j+1/2}$ when $v_{i,j+1/2} > 0$,

$$\tilde{s}_{i,j+1/2} = \tilde{s}_{i,j} + \frac{\Delta z}{2}\left(1 - v_{i,j+1/2}\frac{\Delta t}{\Delta z}\right)\frac{\tilde{s}_{i,j+1} - \tilde{s}_{i,j-1}}{\Delta z} \tag{44}$$

and when $v_{i,j+1/2} < 0$,

$$\tilde{s}_{i,j+1/2} = \tilde{s}_{i+1,j} - \frac{\Delta z}{2}\left(1 + v_{i,j+1/2}\frac{\Delta t}{\Delta z}\right)\frac{\tilde{s}_{i,j+2} - \tilde{s}_{i,j}}{\Delta z}. \tag{45}$$

For the case when s represents the volume-of-fluid function F we have the following representation for $\tilde{s}_{i,j+1/2}$ when $v_{i,j+1/2} > 0$,

$$\tilde{s}_{i,j+1/2} = \frac{\left(\int_{z_{i+1/2}-v_{i,j+1/2}\Delta t}^{z_{i+1/2}} \int_{r_{i-1/2}}^{r_{i+1/2}} H\left(\tilde{\phi}^R_{i,j}(r,z)\right) r \, dr \, dz\right)}{r_i v_{i,j+1/2} \Delta t \Delta r} \tag{46}$$

and when $v_{i,j+1/2} < 0$,

$$\tilde{s}_{i,j+1/2} = \frac{\left(\int_{z_{i+1/2}-v_{i,j+1/2}\Delta t}^{z_{i+1/2}} \int_{r_{i-1/2}}^{r_{i+1/2}} H\left(\tilde{\phi}_{i,j}^{R}(r,z)\right) r \, dr \, dz\right)}{r_i \left|v_{i,j+1/2}\right| \Delta t \, \Delta r}. \tag{47}$$

The linear reconstruction $\tilde{\phi}_{i,j}^{R}(r,z)$ found in (46) and (47) has an analogous form as (38),

$$\tilde{\phi}_{i,j}^{R}(r,z) = \tilde{a}_{i,j}(r - r_i) + \tilde{b}_{i,j}(z - z_j) + \tilde{c}_{i,j}. \tag{48}$$

After ϕ^{n+1} and F^{n+1} have been updated according to (28) and (29) we "couple" the level set function to the volume fractions by assigning the level set function ϕ^{n+1} to be the *exact* signed normal distance to the reconstructed interface. The algorithm to find the signed normal distance in a strip of K cells about the reconstructed interface is as follows:

1. Truncate the volume fractions,

$$F_{i,j}^{n+1} = \begin{cases} 0 & \text{if } F_{i,j}^{n+1} \le 0 \quad \text{or} \quad \phi_{i,j}^{n+1} < -\Delta r \\ 1 & \text{if } F_{i,j}^{n+1} \ge 1 \quad \text{or} \quad \phi_{i,j}^{n+1} > \Delta r \\ F_{i,j}^{n+1} & \text{otherwise.} \end{cases} \tag{49}$$

2. Tag all computational cells (i, j).
3. In each computational cell (i, j), check to see if

$$\phi_{i,j}^{n+1} \phi_{i',j'}^{n+1} \le 0 \tag{50}$$

for some $|i - i'| \le 1$, $|j - j'| \le 1$; if there is a (i', j') such that (50) is satisfied, then perform the following steps:

(a) If

$$0 < F_{i,j}^{n+1} < 1 \tag{51}$$

and

$$\phi_{i,j}^{n+1}\left(\phi_{i,j}^{n+1} + \phi_{i',j'}^{n+1}\right) \le 0 \qquad \text{for some } |i - i'| \le 1, |j - j'| \le 1, \tag{52}$$

then construct the linear reconstruction $\phi_{i,j}^{n+1,R}(r,z)$ (38),

$$\phi_{i,j}^{n+1,R}(r,z) = a_{i,j}(r - r_i) + b_{ij}(z - z_j) + c_{i,j}. \tag{53}$$

If (51) or (52) is not satisfied then mark all of cell (i, j) face centroids and corners as either "positive" or "negative" depending on the sign of $\phi_{i,j}^{n+1}$. If both (51) and (52) are satisfied, mark all of cell (i, j) face centroids and corners according to the sign of $\phi_{i,j}^{n+1,R}(r,z)$ evaluated at the face centroids and corners.

(b) For each cell (i', j'), $(i' - i)^2 + (j' - j)^2 < K^2$ and $(i' - i)^2 + (j' - j)^2 < (|\phi_{i',j'}^{n+1}|/\Delta r + 2)^2$ do the following steps; we refer the reader to the diagram in Fig. 3 in Section 4.

(i) Determine the closest point on the boundary of cell (i, j) to $(r_{i'}, z_{j'})$ (this point will always either be at the corner or face centroid of the cell boundary). If the sign of the level set function at the closest point is *opposite* of $\phi^{n+1}_{i',j'}$, then set d, the shortest distance associated with cells (i, j) and (i', j'), equal to the distance from $(r_{i'}, z_{j'})$ to the closest point on the boundary of cell (i, j). If the sign of the level set function at the closest point is the *same* as $\phi^{n+1}_{i',j'}$ and (51), (52) are both satisfied, then let d be the shortest distance between $(r_{i'}, z_{j'})$ and the line segment represented by $\phi^{n+1,R}_{i,j}(r, z)$.

(ii) Update $\phi^{n+1}_{i',j'}$ using d:

$$\phi^{n+1}_{i',j'} = \begin{cases} \operatorname{sign}\left(\phi^{n+1}_{i',j'}\right) d & \text{if } d < \left|\phi^{n+1}_{i',j'}\right| \text{ or cell } (i', j') \text{ is tagged} \\ \phi^{n+1}_{i',j'} & \text{otherwise.} \end{cases}$$

(iii) Untag cell (i', j').

4. For cells (i, j) which are still tagged, we have

$$\phi^{n+1}_{i,j} = \begin{cases} -K\Delta r - \Delta r & \text{if } \phi^{n+1}_{i,j} < 0 \\ K\Delta r + \Delta r & \text{if } \phi^{n+1}_{i,j} > 0. \end{cases} \tag{54}$$

Remarks.

• The coupling between the level set function ϕ and the volume-of-fluid function F occurs when computing the normal of the reconstructed interface (38) and also when assigning the level set function with the exact signed normal distance to the reconstructed interface.

• In order to find the shortest distance between the cell center (i', j') and the line segment represented by $\phi^{n+1,R}_{i,j}(r, z)$ (53), one first re-scales (53) so that $a^2_{i,j} + b^2_{i,j} = 1$. The distance is then $d = \phi^{n+1,R}_{i,j}(r_{i'}, z_{j'})$. The point $\boldsymbol{x}^c = (r_{i'}, z_{j'}) - d\nabla\phi^{n+1,R}$ is the point where the normal extension from (i', j') to $\phi^{n+1,R}_{i,j}(r, z)$ intersects $\phi^{n+1,R}_{i,j}(r, z)$. If $\boldsymbol{x}^c$ falls outside of cell (i, j), then the shortest distance between (i', j') and $\phi^{n+1,R}_{i,j}(r, z)$ must be the distance from (i', j') to one of the end points of the line *segment* represented by $\phi^{n+1,R}_{i,j}(r, z)$. In three dimensions, it becomes only slightly more complicated. The step for finding the normal distance to the plane represented by $\phi^{n+1,R}_{i,j,k}(x, y, z)$ is simply $d = \phi^{n+1,R}_{i,j,k}(x_{i'}, y_{j'}, z_{k'})$ (assuming the coefficients are appropriately scaled). As for the axisymmetric case, $\boldsymbol{x}^c = (x_{i'}, y_{j'}, z_{k'}) - d\nabla\phi^{n+1,R}$. The only added complication in three dimensions is finding the shortest distance to the end points of the reconstructed plane. But this can be handled by projecting the point $(x_{i'}, y_{j'}, z_{k'})$ onto the plane that coincides with each of cell (i, j, k)'s faces and then finding the distance from the projected point to the intersection between the reconstructed plane and the cell face. The procedure in this case reduces to the procedure for the 2d case.

• One difference between our algorithm and the one presented by Bourlioux [10] for coupling ϕ to F is in how the new level set function ϕ^{n+1} is reinitialized using the volume-of-fluid function F^{n+1}. In [10], the level set function ϕ^{n+1} is modified to be the intercept $c_{i,j}$ (38) of the reconstructed interface $\phi^{n,R}_{i,j}(r, z)$ in cells that contain a piecewise linear reconstruction. For cells that do not contain the piecewise linear reconstruction, Bourlioux used the redistance iteration proposed by [33]. In our computations we have tried both approaches and found more accurate results when assigning ϕ^{n+1} to be the *exact* distance from the reconstructed interface. In our implementation, we only need ϕ^{n+1} to be the exact

distance within K cells where $K < 5$ (K is dependent on the "interface thickness," see Section 5). The speed for our algorithm is thus $O(K^2 N)$ where N is the number of computational cells that contain a volume-of-fluid reconstruction. This is the same speed estimate as for a redistance iteration within a tube K cells thick about the zero level set.

- In the previous remark, it was recommended that one find the exact distance to the reconstructed interface when implementing the CLSVOF algorithm. It is *not* recommended that one reconstruct the interface if one is only using the LS approach for moving the free surface. In this case, the redistance iteration [30, 31] is recommended since interface reconstructions using only the level set function without the aid of volume fractions will incur considerably more error than the iteration approach outlined in [30, 31].

4. TEMPORAL DISCRETIZATION

Our discretization procedure for approximating (7) is based on the variable density projection method described by Bell *et al.* [5], Bell and Marcus [7], Almgren *et al.* [4], and Puckett *et al.* [23]. The discrete velocity field $\boldsymbol{U}^n_{i,j}$, level set function $\phi^n_{i,j}$, and volume fractions $F^n_{i,j}$ are located at cell centers. The pressure $p^{n-1/2}_{i+1/2,j+1/2}$ is located at cell corners. A diagram of where the discrete variables are located in relation to the computational grid is shown in Fig. 2. J represents the index of the computational cell closest to the top physical boundary.

The time stepping procedure is based on the Crank–Nicholson method. At the beginning of each time step, we are given the velocity $\boldsymbol{U}^n$, level set function ϕ^n, and volume fractions F^n at time t^n. We are also given the lagged pressure gradient $p^{n-1/2}$. The density $\rho^n = \rho(\phi^n)$, viscosity $\mu^n = \mu(\phi^n)$, and Heaviside function $H^n = H(\phi^n)$ are given at time t^n since they are functions of ϕ^n.

We discretize (7) and (9) in time using the steps below; for details of the spatial discretization of the nonlinear terms, viscous terms, surface tension terms, and projection step, see the Appendix (Section 9).

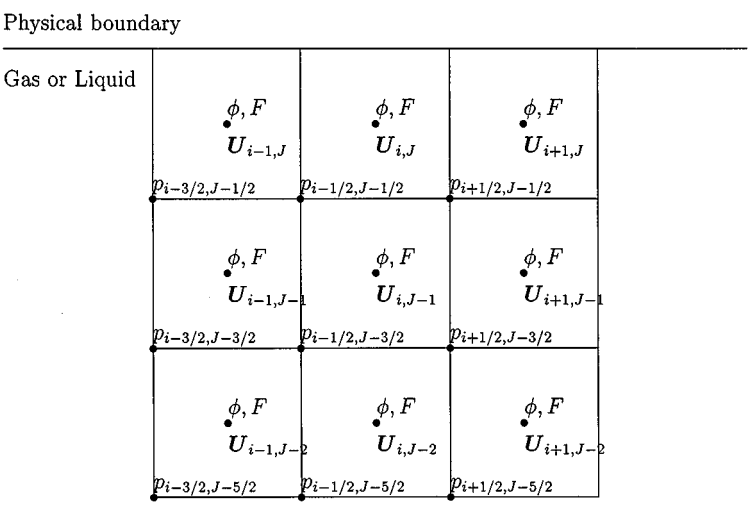

FIG. 2. Diagram of where the discrete variables $\boldsymbol{U}$, p, ϕ, and F are located in relation to the computational grid and the physical boundary.

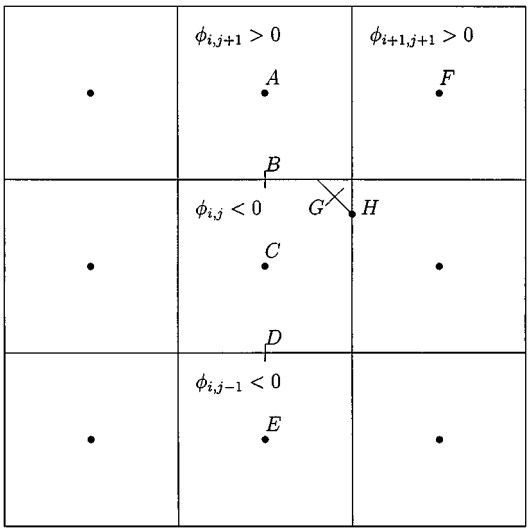

FIG. 3. The distances d assigned to points A, F, and E respectively are $|\overline{AB}|$, $|\overline{FG}|$, and $|\overline{EH}|$.

1. Predict the edge based velocity field $u^{ADV,n+1/2}_{i+1/2,j}$ and $v^{ADV,n+1/2}_{i,j+1/2}$ using an explicit predictor-corrector scheme. The edge based velocity field shall be discretely divergence-free; i.e., they satisfy (23). A description of how $u^{ADV,n+1/2}_{i+1/2,j}$ and $v^{ADV,n+1/2}_{i,j+1/2}$ are formed is described in the Appendix (Subsection 9.1).

2. Given ϕ^n, F^n, and $\boldsymbol{U}^{ADV,n+1/2}$, apply the CLSVOF advection algorithm described in Section 3 in order to determine ϕ^{n+1} and F^{n+1}.

Once ϕ^{n+1} is obtained, the following quantities are updated,

$$\phi^{n+1/2} = \frac{1}{2}(\phi^n + \phi^{n+1}) \tag{55}$$

$$\rho^{n+1/2} = \rho\left(\phi^{n+1/2}\right) \tag{56}$$

$$\mu^{n+1/2} = \mu\left(\phi^{n+1/2}\right). \tag{57}$$

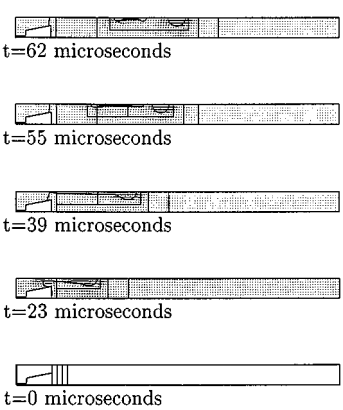

FIG. 4. Axisymmetric jetting of ink. $\rho_w/\rho_a = 816$, $\mu_w/\mu_a = 64$. Effective fine grid resolution is 64×1024.

3. Semi-implicit viscous solve for the intermediate velocity $\boldsymbol{U}^*$,

$$\frac{\boldsymbol{U}^* - \boldsymbol{U}^n}{\Delta t} = -[(\boldsymbol{U}\cdot\nabla)\boldsymbol{U}]^{n+\frac{1}{2}} - \frac{Gp^{n-\frac{1}{2}}}{\rho^{n+\frac{1}{2}}} + \frac{\mathcal{L}^* + \mathcal{L}^n}{2\rho^{n+\frac{1}{2}}} - \frac{\mathcal{M}^{n+1/2}}{\rho^{n+\frac{1}{2}}} + \boldsymbol{F}. \tag{58}$$

$\mathcal{L}$ is a second-order finite difference approximation to $\nabla\cdot(2\mu(\phi)\mathcal{D})$, $\mathcal{M}$ is a finite difference approximation to $\gamma\kappa(\phi)\nabla H$, and Gp is an approximation to ∇p. In the Appendix (Subsection 9.2) we give a description of Gp, $\mathcal{L}$, and $\mathcal{M}$. The nonlinear advection term $[(\boldsymbol{U}\cdot\nabla)\boldsymbol{U}]^{n+1/2}$ is evaluated using an explicit predictor-corrector scheme and requires only the available data at t^n. In the Appendix (Subsection 9.1), we give a description of how $[(\boldsymbol{U}\cdot\nabla)\boldsymbol{U}]^{n+1/2}$ is discretized. The density ρ, viscosity μ, Heaviside function H, and curvature κ are constructed from the level set function calculated at time $t^{n+1/2}$ in the level set advection step (55). The lagged pressure gradient $Gp^{n-1/2}$ and force $\boldsymbol{F}$ are treated as source terms.

Equation (58) when discretized results in a coupled parabolic solve for all velocity components of $\boldsymbol{U}^*$. We use multigrid as an iteration method for solving (58).

4. Projection step for $\boldsymbol{U}^{n+1}$,

$$\begin{aligned}\frac{\boldsymbol{U}^{n+1} - \boldsymbol{U}^n}{\Delta t} &= \mathcal{P}\left(\frac{\boldsymbol{U}^* - \boldsymbol{U}^n}{\Delta t}\right)\\ \frac{1}{\rho^{n+\frac{1}{2}}}Gp^{n+\frac{1}{2}} &= \frac{1}{\rho^{n+\frac{1}{2}}}Gp^{n-\frac{1}{2}} + (\boldsymbol{I}-\mathcal{P})\left(\frac{\boldsymbol{U}^* - \boldsymbol{U}^n}{\Delta t}\right).\end{aligned} \tag{59}$$

$\mathcal{P}$ represents the discretization of the projection operator (21). In the Appendix (Subsection 9.3) we give a description of $\mathcal{P}$.

4.1. Time step. The time step Δt at time t^n is determined by restrictions due to the CFL condition, gravity, viscosity, and surface tension [11, 33],

$$\Delta t < \min_{i,j}\left(\sqrt{\frac{(\rho_1+\rho_2)}{8\pi\gamma}}\Delta r^{3/2},\ \frac{3}{14}\frac{\rho^n\Delta r^2}{\mu^n},\ \frac{\Delta r}{|\boldsymbol{u}^n|},\ \frac{2\Delta r}{|\boldsymbol{u}^n| + \sqrt{|\boldsymbol{u}^n|^2 + 4\mathcal{F}^n\Delta r}}\right),$$

where

$$\mathcal{F}^n = \left|-\frac{Gp^{n-1/2}}{\rho^n} + \frac{\mathcal{L}^n}{\rho^n} - \frac{\mathcal{M}^n}{\rho^n} + \boldsymbol{F}\right|.$$

The last time step constraint is justified through the following simplified analysis. If we consider the simplified equation,

$$\begin{aligned}u_t &= F\\ u(t_n) &= u_n\end{aligned}$$

then the solution at t_{n+1} is

$$u(t_{n+1}) = u_n + \Delta t F.$$

We require a "CFL" type condition,

$$u(t_{n+1})\Delta t < \Delta r.$$

The resulting equation for Δt is

$$(u_n + \Delta t F)\Delta t < \Delta r.$$

5. INTERFACE THICKNESS

We shall give the interface a numerical thickness as was done in [33, 35]. Numerically, we substitute the smoothed Heaviside function $H_\epsilon(\phi)$ for the sharp Heaviside function $H(\phi)$. The smoothed Heaviside function is defined as

$$H_\epsilon(\phi) = \begin{cases} 0 & \text{if } \phi < -\epsilon \\ \frac{1}{2}\left[1 + \frac{\phi}{\epsilon} + \frac{1}{\pi}\sin(\pi\phi/\epsilon)\right] & \text{if } |\phi| \le \epsilon \\ 1 & \text{if } \phi > \epsilon. \end{cases} \tag{60}$$

Assume that ϕ represents the signed normal distance to the free surface. By giving the interface a thickness of 2ϵ we eliminate problems when solving (91) and also when discretizing the surface tension term

$$\gamma \frac{\kappa(\phi)\nabla H(\phi)}{\rho(\phi)}.$$

In our algorithm, the front will have a uniform thickness since the level set function ϕ always represents the signed distance to the free surface due to our CLSVOF advection algorithm (see Section 3).

6. RESULTS

In this section we test the CLSVOF method on 3D and axisymmetric problems in which surface tension effects and changes in topology are present. We compare the CLSVOF method with the LS method [32, 33], the VOF method [1, 22], and the boundary integral method [32]. We shall show that our coupled level set volume-of-fluid method (CLSVOF) has comparable accuracy to the LS method for computing most surface tension driven flows. We also show that our CLSVOF method conserves mass to within a fraction of a percent for all of our test cases.

6.1. Validation of the CLSVOF advection algorithm. We consider two problems in this section; the translation of a circle in a periodic domain and the rotation of a notched disk (Zalesak's problem [36]). These problems are 2D problems and not axisymmetric problems.

For the translation of a circle, we have a 4×4 periodic domain and a unit circle initialized at the center of the domain. The constant velocity field is given by $u = 1$ and $v = 0$. In Tables I and II, we display the error at $t = 4$ for grid resolutions ranging from 32×32 to 256×256.

TABLE I
The Error $E(t = 1)$ after Translating a Unit Circle One Unit in Time with CFL Number 1

Δx	CLSVOF	ELVIRA
1/8	0.000523	0.000610
1/16	0.000128	0.000160
1/32	0.000031	0.000040
1/64	0.000008	0.000010

TABLE II
The Error $E(t = 1)$ after Translating a Unit Circle One Unit in Time with CFL Number 1/32

Δx	CLSVOF	ELVIRA
1/8	0.001110	0.001380
1/16	0.000327	0.000437
1/32	0.000097	0.000125
1/64	0.000029	0.000038

These errors are compared to the errors reported in [22] when using the ELVIRA volume-of-fluid interface reconstruction algorithm. We measured the error in the following way

$$E(t) = \frac{1}{L} \sum_{i,j} \int_{\Omega_{ij}} |H(\phi_e(t)) - H(\phi_c(t))| \, d\boldsymbol{x}, \tag{61}$$

where ϕ_e is the level set function representing the exact solution and ϕ_c is the level set function representing the computed solution. L is the perimeter of the interface. The integral in (61) is approximated by partitioning Ω_{ij} into 128×128 rectangles and then applying the midpoint rule. The values of ϕ_e and ϕ_c at the midpoint of each rectangle are obtained via bilinear interpolation.

In spite of the fact that we truncate the volume fractions which are farther than Δx from the zero level set, the maximum mass fluctuation for the translating circle problem is $1.0E-9$. The mass is measured as

$$V(t) = \sum_{i,j} F_{i,j} \Delta r \Delta z. \tag{62}$$

For the rotation of a notched disc, we have a 1×1 periodic domain and a notched circle of radius 0.15 units positioned at (0.5, 0.75). The width of the notch is 0.05 units and the height of the notch is 0.25 units. The (stationary) velocity field is given by

$$u = \frac{\pi}{3.14}(0.5 - y) \tag{63}$$

$$v = \frac{\pi}{3.14}(x - 0.5). \tag{64}$$

In Table III we display the error $E(628)$ for grid resolutions ranging from 100×100 to 400×400. These errors are compared to the errors reported in [22] when using the ELVIRA method for volume-of-fluid reconstruction. In Fig. 5 we display the interface profile for the 100×100 computation.

TABLE III
The Error $E(628.0)$ for Zalesak's Test Problem

Δx	CLSVOF	ELVIRA
1/16	0.00572	0.00567
1/32	0.00252	0.00262
1/64	0.00106	0.00121

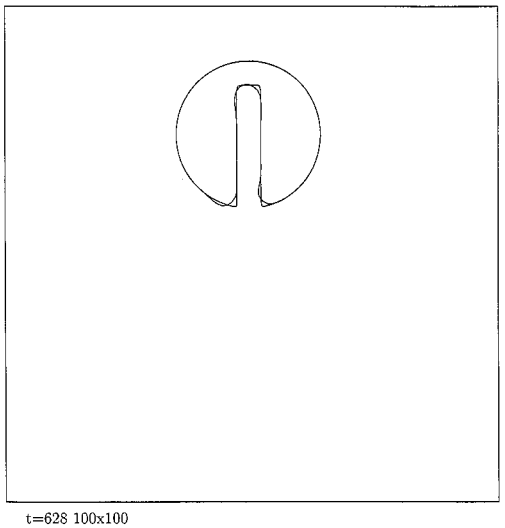

FIG. 5. Comparison with expected answer for Zalesak's problem after one full revolution of notched disk. Grid resolution is 100×100.

6.2. Zero-gravity drop oscillation. We compute axisymmetric zero-gravity drop dynamics using the CLSVOF method and compare our results to those using the LS method. According to the linearized results derived by Lamb [19, Sect. 275], the position of the drop interface is

$$R(\theta, t) = a + \epsilon P_n(\cos(\theta)) \sin(\omega_n t),$$

where

$$\omega_n^2 = \gamma \frac{n(n-1)(n+1)(n+2)}{a^3(\rho_l(n+1) + \rho_g n)}$$

and P_n is the Legendre polynomial of order n. θ runs between 0 and 2π where $\theta = 0$ corresponds to $r = 0$ and $z = a$. If viscosity is present, Lamb [19, Sect. 355] found that the amplitude is proportional to $e^{-t/\tau}$, where

$$\tau = \frac{a^2 \rho_l}{\mu_l (2n+1)(n-1)}.$$

We compute the evolution of a drop with $a = 1$, $g = 0$, $\mu_l = 1/200$, $\mu_l/\mu_g = 100$, $\gamma = 1/2$, $\rho_l = 1$, and $\rho_l/\rho_g = 100$. The initial interface is given by $R(\theta, \pi/2)$, with $\epsilon = 0.05$ and $n = 2$. With these parameters we find $\omega_2 = 2.00$ and $\tau = 38.3$. The fluid domain is $\Omega = \{(r, z) \mid 0 \le r \le 1.5 \text{ and } 0 \le z \le 1.5\}$ and we compute on grid sizes ranging from 32×32 to 128×128. Symmetric boundary conditions are imposed at $r = 0$ and $z = 0$. The interfacial thickness parameter α is two grid cells. The results of our computations using the CLSVOF method are compared to those of the LS method in Fig. 6 where we display the perturbation in the major axis on a 128×128 grid. The period is 3.17 and the expected linearized period is π. In Table IV, we display the relative error between succeeding resolutions for the major amplitude of the droplet. The major amplitude for a grid cell size h, $R_h(\pi/2, t)$, is measured

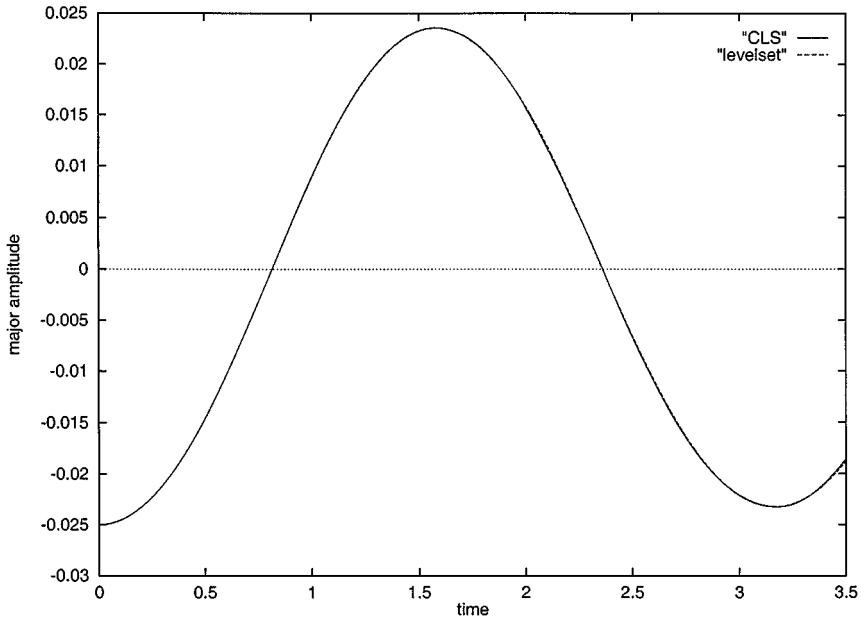

FIG. 6. perturbation in major amplitude for zero gravity drop oscillations. $\mu_l = 1/200$, $\gamma = 1/2$, density ratio 100 : 1, viscosity ratio 100 : 1. Comparison of coupled LS method (CLSVOF) with the LS method.

by constructing a piecewise linear interpolant of the level set function along the horizontal axis and determining the position of the zero crossing of the piecewise linear interpolant. Then we define the L^1 error by

$$\text{err}_1 \equiv \int_0^{\pi} |R_h(\pi/2, t) - R_{2h}(\pi/2, t)|\, dt,$$

and the L^∞ error by

$$\text{err}_\infty \equiv \max_{0 \le t \le \pi} |R_h(\pi/2, t) - R_{2h}(\pi/2, t)|.$$

In order to compute the error numerically, we subdivide the time interval into n equally spaced intervals where n is the number of time steps at the fine grid resolution specified by h. The values on the coarse grid are interpolated in time and then compared with that on the fine grid.

TABLE IV
Convergence Study Using the CLSVOF Algorithm for Zero Gravity Drop Oscillations $\gamma = 1/2$, $\mu_l = 1/200$, $\mu_l/\mu_g = 100$, $\rho_l/\rho_g = 100$, and $\alpha = 2$

Δr	L_1	L_∞	period
3/64	N/A	N/A	3.18
3/128	0.00029	0.00066	3.17
3/256	0.00012	0.00032	3.17

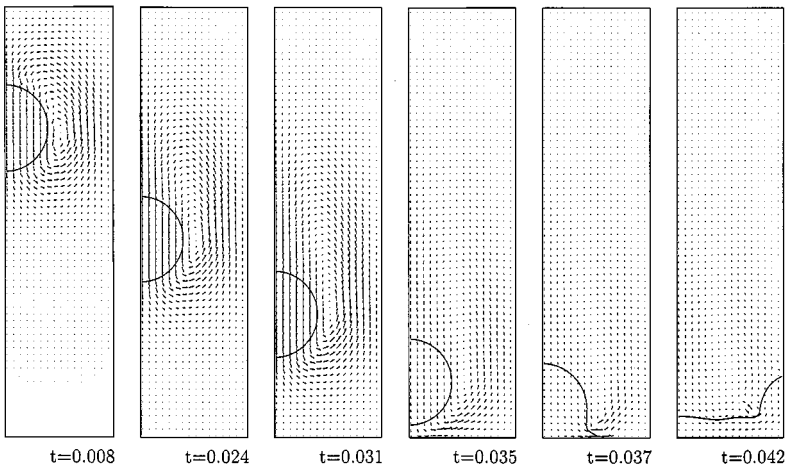

FIG. 7. Falling 1-mm 2D water drop in air; density ratio 816 : 1, viscosity ratio 64 : 1, number of grid cells 32×128.

6.3. Drop impacting solid wall. Here we test the CLSVOF method on the 2D falling drop problem. The parameters are the same as that computed by Aleinov and Puckett [1] except that we compute with a density ratio of 816 : 1 as opposed to a density ratio of 10 : 1. Also, the viscosity ratio in our computations is 64 : 1 as opposed to constant viscosity. The radius of the falling drop is 0.1 cm. Other relevant parameters are $g = 980$ cm/s^2, $\sigma = 73.2$ g/s^2, $\mu_g = 1.78E-4$ g/(cms), $\mu_l = 1.137E-2$ g/(cms), $\rho_g = 0.001225$ g/cm^3, and $\rho_l = 1.0$ g/cm^3. The dimension of our domain is 0.25 by 1.0 cm. In Fig. 7 we show the evolution of the drop as it hits the bottom of our domain (mesh size 32×128).

In Table V, we show the relative errors in the position of the interface and the velocity ($t = 0.025$) for grid resolutions ranging from 16×64 to 64×256. The error for the position of the interface is measured similarly as in (61) except that we measure the relative error between succeeding grid resolutions since we do not know the exact solution for this problem,

$$E(t) = \sum_{i,j} \int_{\Omega_{ij}} |H(\phi_f(t)) - H(\phi_c(t))| \, d\boldsymbol{x}. \tag{65}$$

Here, ϕ_c is the level set function from a coarser computation and ϕ_f is the level set function

TABLE V
Convergence Study at $t = 0.025$ for Falling 2D Drop of Water in Air

Δx	$E(0.025)$	Center of mass	$E_{u,L1}(0.025)$	$E_{u,\max}(0.025)$
1 cm/64	N/A	0.4372	N/A	N/A
1 cm/128	0.00202	0.4382	5.34	2.40
1 cm/256	0.00058	0.4379	2.76	1.72

Note. Radius of drop is 0.1 cm. The density ratio is 816 : 1 and the viscosity ratio is 64 : 1. Results computed using the CLSVOF method $\alpha = 2$.

from the refined computation. The relative error for the velocity is measured by the equations

$$E_{u,L1}(t) = \sum_{i,j} \sqrt{(u_{f,ij} - u_{c,ij})^2 + (v_{f,ij} - v_{c,ij})^2}\,\Delta x\,\Delta y \tag{66}$$

$$E_{u,\max}(t) = \max_{i,j} \sqrt{(u_{f,ij} - u_{c,ij})^2 + (v_{f,ij} - v_{c,ij})^2}. \tag{67}$$

In Table V, we also display the position of the center of mass at $t = 0.025$.

6.4. Capillary instability. As mentioned previously, one motivation for our development of the CLSVOF method stems from the modeling of micro-scale jetting devices (e.g., see [2, 16]). Typically, when a fluid is ejected from such a device, a long tail of liquid forms behind a roughly spherical drop. Typically this tail undergoes a capillary instability which causes it to break up into droplets. The ability to accurately model surface tension is critically important in order to accurately model the dynamics of this process on a micro-scale. In this section, we test the CLSVOF method on the classical Rayleigh capillary instability problem in which a slightly perturbed cylindrical column of liquid is driven to break up into droplets by surface tension (capillary) effects. In this test problem we use parameters that are comparable to those found in problems involving micro-scale jetting devices.

We consider an initially perturbed cylindrical column of water in air. The shape of the initial interface is

$$r(z) = r_0 + \epsilon \cos(2\pi z/\lambda). \tag{68}$$

We compute on an axisymmetric domain $\Omega = \{(r, z) \mid 0 \le r \le \lambda/4 \text{ and } 0 \le z \le \lambda/2\}$. Symmetric boundary conditions are enforced at $r = 0$, $z = 0$, and $z = \lambda/2$. Outflow boundary conditions are enforced at $r = \lambda/4$. The relevant dimensional parameters for this test problem are $r_0 = 6.52$ microns, $\epsilon = 1.3$ microns, $\lambda = 60$ microns, $u_l = 1.138 \times 10^{-2}$ g/(cms), $\mu_g = 1.77 \times 10^{-4}$ g/(cms), $\rho_l = 1.0$ g/cm^3, $\rho_g = 0.001225$ g/cm^3, and $\gamma = 72.8$ dynes/cm. In our computations we use the following dimensionless parameters: the Reynolds number $R = \rho_l L U/\mu_l = 7.5$, the Weber number $W = \rho_l L U^2/\gamma = 11$, $L = 1$ micron, $U = 8.53$ m/s, and the density and viscosity ratios are 816 and 64, respectively.

In Fig. 8, we display the results of our computations using the CLSVOF method for the capillary jet as it breaks up. As a comparison, we have also displayed computations using the LS approach (see Fig. 9). In Tables VI and VII, we measure the relative errors for the interface and velocity field for grid resolutions ranging from 32×64 to 128×256. The relative error for the interface is measured by (65). The relative error for the velocity is measured by the equations

$$E_{u,L1}(t) = \sum_{i,j} \sqrt{(u_{f,ij} - u_{c,ij})^2 + (v_{f,ij} - v_{c,ij})^2}\, r_i\,\Delta r\,\Delta z \tag{69}$$

$$E_{u,\max}(t) = \max_{i,j} \sqrt{(u_{f,ij} - u_{c,ij})^2 + (v_{f,ij} - v_{c,ij})^2}. \tag{70}$$

As shown in the tables, we obtain about first order accuracy before and after pinch off. We suspect that we have only first order accuracy for this problem because the density and viscosity jump across the interface. We recomputed the capillary jet problem with constant density $\rho_l = \rho_g = 1.0$ g/cm^3 and constant viscosity $\mu_l = \mu_g = 1.138 \times 10^{-2}$ g/(cms) and the

TABLE VI

The Convergence of the Numerical Solution Obtained with the CLSVOF Algorithm to the Rayleigh Capillary Instability Problem with a Jump in Density and Viscosity $\mu_l/\mu_g = 64$ and $\rho_l/\rho_g = 816$ at Time $t = 80$

Grid	$E(80)$	$E_{u,L1}(80)$	$E_{u,\max}(80)$
16×32	N/A	N/A	N/A
32×64	36.88	268.7	0.064
64×128	21.13	167.5	0.050
128×256	8.14	94.2	0.034

Note. This is *before* the interface has broken up into droplets and the rate of convergence is roughly first-order, except in the sup (i.e., max) norm which does not appear to decrease with decreasing $\Delta x = \Delta y$.

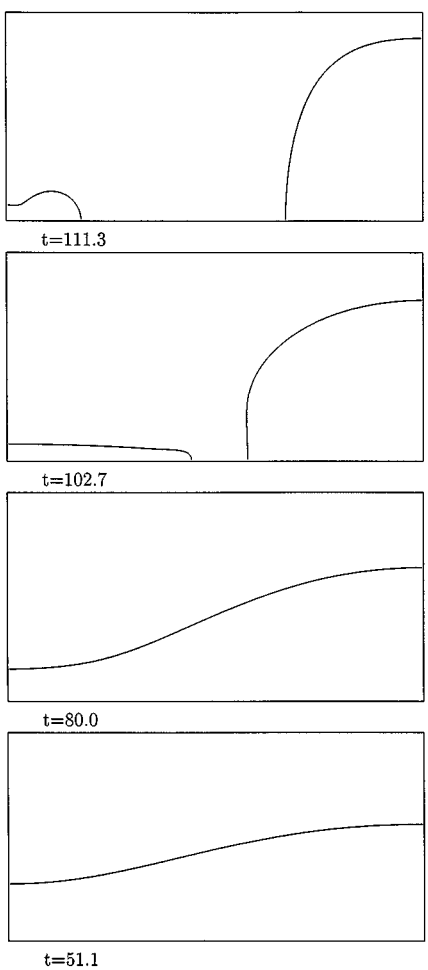

FIG. 8. Capillary instability. $\rho_w/\rho_a = 816$, $\mu_w/\mu_a = 64$. Grid resolution is 64×128.

TABLE VII

The Convergence of the Numerical Solution Obtained with the CLSVOF Algorithm to the Rayleigh Capillary Instability Problem with a Jump in Density and Viscosity $\mu_l/\mu_g = 64$ and $\rho_l/\rho_g = 816$ at Time $t = 120$

Grid	$E(120)$	$E_{u,L1}(120)$	$E_{u,\max}(120)$
16×32	N/A	N/A	N/A
32×64	318.9	936.2	0.90
64×128	182.9	599.3	1.32
128×256	76.8	150.6	0.53

Note. This is *after* the interface has broken up into droplets yet the rate of convergence is still roughly first-order, except in the sup (i.e., max) norm.

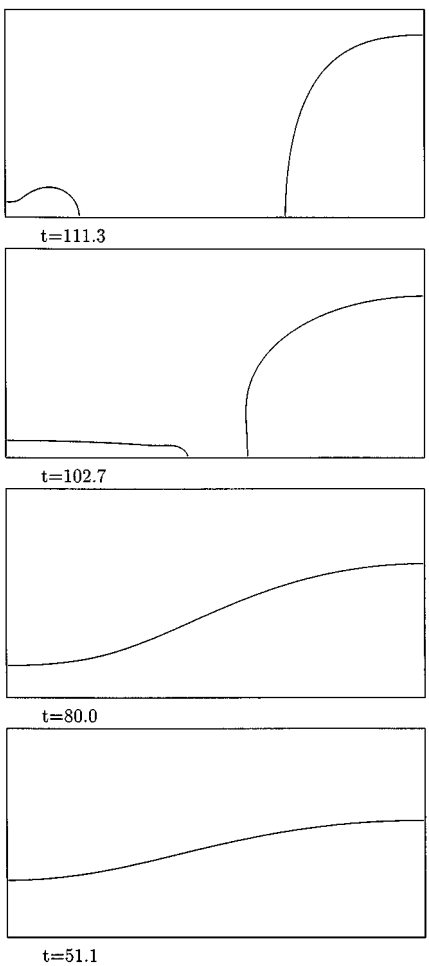

FIG. 9. Capillary instability; LS method. $\rho_w/\rho_a = 816$, $\mu_w/\mu_a = 64$. Grid resolution is 64×128.

TABLE VIII

The Convergence of the Numerical Solution Obtained with the CLSVOF Algorithm to the Rayleigh Capillary Instability Problem with No Jump in the Density or Viscosity at the Interface $\mu_l/\mu_g = 1$, $\rho_l/\rho_g = 1$ at Time $t = 120$

Grid	$E(120)$	$E_{u,L1}(120)$	$E_{u,\max}(120)$
16×32	N/A	N/A	N/A
32×64	52.76	68.55	0.1535
64×128	11.23	16.17	0.0339
128×256	2.82	4.24	0.0091

Note. Now the rate of convergence is roughly second-order, except for the error in the sup (i.e., max) norm which now exhibits a first-order decrease.

relative errors for the recomputed problem are shown in Tables VIII and IX. The interface profiles for the constant density case are shown in Fig. 10.

6.5. Inviscid gas bubble. We consider the rise of an inviscid axisymmetric gas bubble in liquid. Here, the dimensionless parameters are $Fr = 1$ and $We = 200$. The density ratio is $1:816$. We compute on an axisymmetric domain $\Omega = \{(r, z) \mid 0 \le r \le 3 \text{ and } 0 \le z \le 6\}$. The interfacial thickness spreading parameter α is 3 grid cells. In Fig. 11, we display results computed using the CLSVOF method (thin lines) and compare them with the boundary integral method [32] (thick lines). As a note, in the boundary integral method, the density in the gas is assumed zero and the domain of computation is assumed to have infinite extent whereas in the CLSVOF method, the density ratio is $1:816$ and we use far-field boundary condition; i.e., the pressure on the walls is $p = z/Fr$. The grid resolution for the CLSVOF results is 128×256 whereas 240 points are used to discretize the interface for the boundary integral method.

In Table X we display the relative errors for the inviscid rising bubble problem when the grid is successively refined. The error is computed at $t = 1.3$, when the jet is about ready to break through the top of the bubble.

TABLE IX

The Convergence of the Numerical Solution Obtained with the CLSVOF Algorithm to the Rayleigh Capillary Instability Problem with No Jump in the Density or Viscosity at the Interface $\mu_l/\mu_g = 1$, $\rho_l/\rho_g = 1$ at Time $t = 160$

Grid	$E(160)$	$E_{u,L1}(160)$	$E_{u,\max}(160)$
16×32	N/A	N/A	N/A
32×64	78.3	125.3	0.143
64×128	15.3	47.6	0.177
128×256	3.0	12.9	0.090

Note. This is *after* interface undergoes a topological change and breaks up into droplets, and the rate of convergence still appears to be second-order in $E(t)$ and $E_{u,L1}(160)$ but $E_{u,\max}(t)$ no longer appears to decrease with decreasing $\Delta x = \Delta y$.

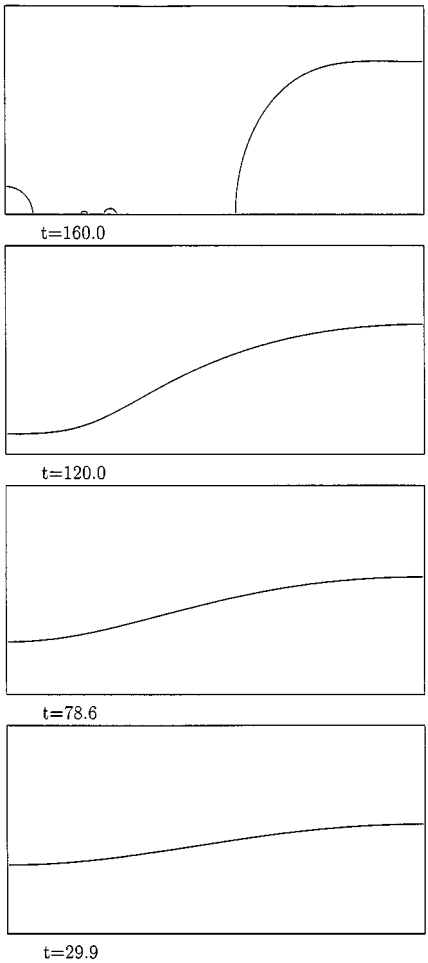

FIG. 10. Capillary instability. $\rho_w/\rho_a = 1$, $\mu_w/\mu_a = 1$. Grid resolution is 64×128.

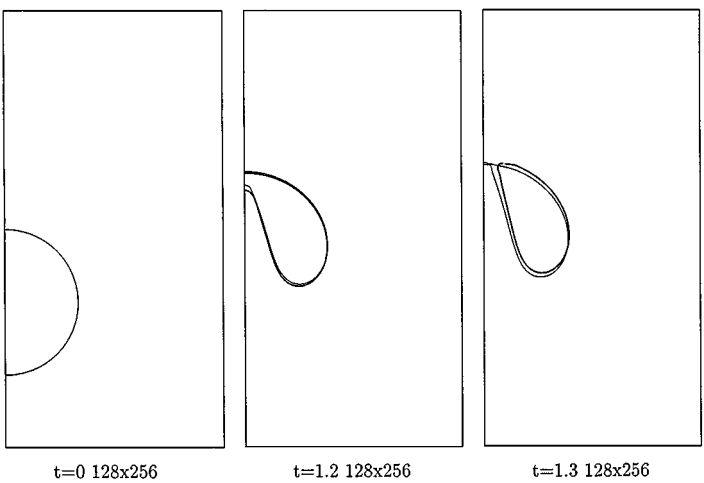

FIG. 11. Spherical gas bubble in liquid. Density ratio 816 : 1, We = 200. Results computed using the volume-of-fluid method (thin contour) are compared to results computed using the boundary integral method (thick contour).

TABLE X
Convergence Study (CLSVOF) for Axisymmetric Inviscid Rising Air Bubble in Water

Δx	$E(1.3)$	$E_{u,L1}(1.3)$	$E_{u,\max}(1.3)$
32×64	N/A	N/A	N/A
64×128	0.167	3.39	2.09
128×256	0.049	2.06	2.34
256×512	0.016	1.19	3.76

Note. The density ratio is 1 : 816. $Fr = 1$, $We = 200$.

In Fig. 12, we overlay results using the LS method (thick line) with those of the CLSVOF method (thin lines) at $t = 1.3$. Here, the results are almost identical.

We have also compared the amount of CPU time each method needs to update the location of the interface in this problem. In the LS computation it took about 3% of the total CPU time to advect the level set function ϕ whereas in the CLSVOF computation it took approximately 10% of the total CPU time for the operator split advection. However, in the LS computation the redistance procedure required 9% of the total CPU time whereas in the CLSVOF computation the redistance procedure only took 2.5% of the total CPU time. Therefore the overall cost to update the location of the interface is comparable.

6.6. Steady rising gas bubble. We compute the steady rise of an axisymmetric gas bubble rising in a viscous liquid. For this problem, the density ratio is 714 : 1 and the viscosity ratio is 6667 : 1. The relevant (dimensionless) parameters for this problem are

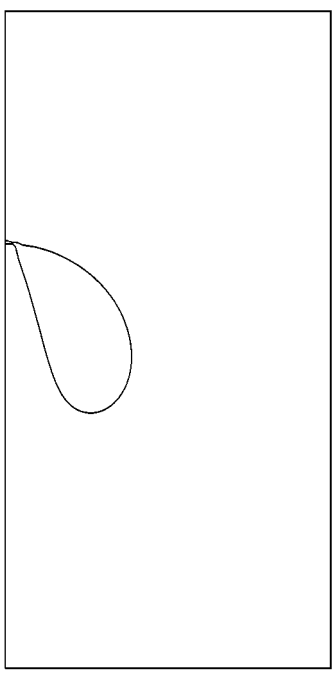

FIG. 12. Spherical gas bubble in liquid. Density ratio 816 : 1, We = 200. Results computed using the volume-of-fluid method (thin contour) are compared to results computed using the LS method (thick contour).

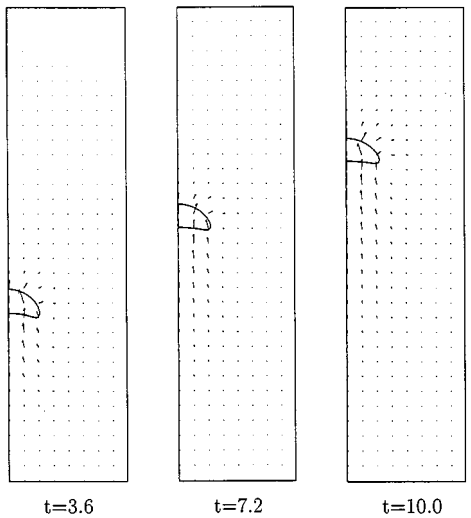

FIG. 13. Rise of an initially spherical gas bubble in viscous liquid. Results computed using the CLSVOF method. Density ratio 714 : 1, viscosity ratio 6667 : 1, $Re = 9.7$, $We = 7.6$, $Fr = 0.78$.

$Fr = 0.78$, $Re = 9.7$, $\mu_l/\mu_g = 6667$, $\rho_l/\rho_g = 714$, and $We = 7.6$. These parameters correspond to those used in bubble experiments by Hnat and Buckmaster [18] and used in steady bubble computations by Ryskin and Leal [26, 27]. Our computational domain is 5×20 and we use a far-field boundary condition (pressure on the walls is $p = z/Fr$). We discretize our domain with 64×256 grid cells. The interfacial thickness parameter α is two grid cells. In Fig. 13, we show the results using the CLSVOF method. In Fig. 14, we plot the position of

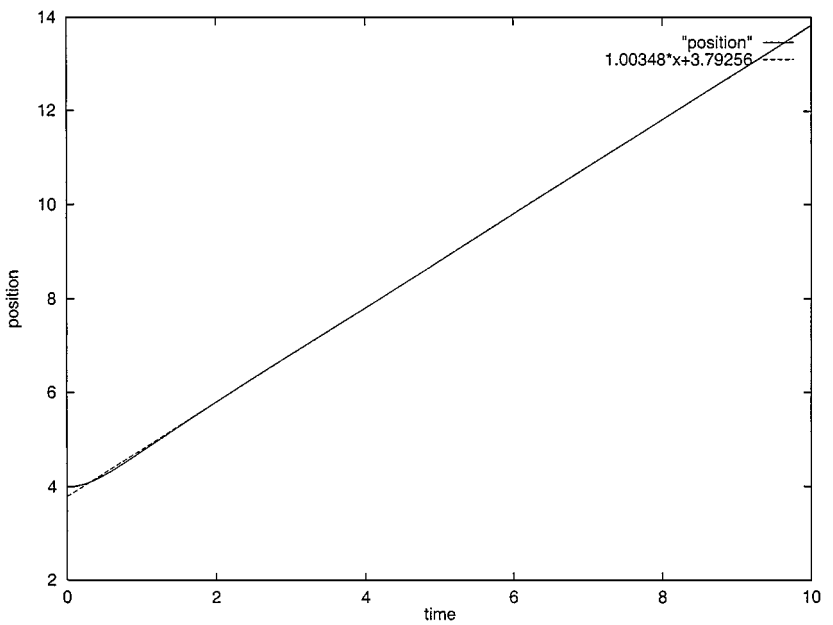

FIG. 14. Comparison of position of center of mass of rising viscous gas bubble in liquid to "best fit" straight line of the position. Expected slope as predicted by experiments is 1. Results computed using the CLSVOF method. Density ratio 714 : 1, viscosity ratio 6667 : 1, $Re = 9.7$, $We = 7.6$, $Fr = 0.78$.

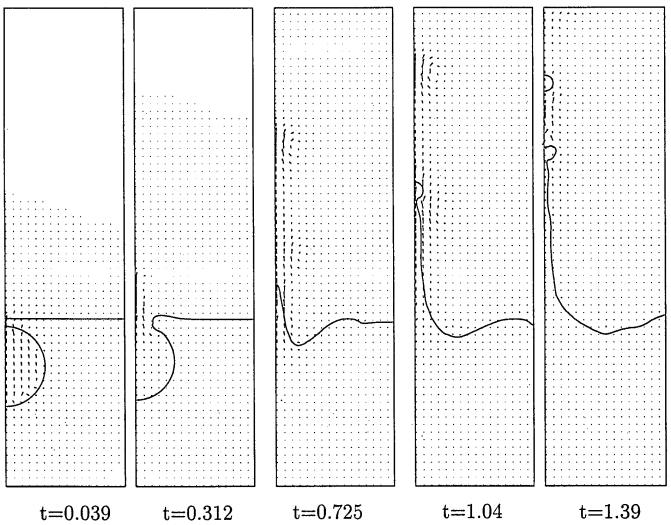

FIG. 15. Evolution of 4-mm air bubble rising to an air/water interface. $\mu_l/\mu_g = 100$, $\rho_l/\rho_g = 1000$, $Fr = 0.64$, $Re = 474$, and $We = 1.0$. Results computed using CLSVOF method.

the center of mass versus time. Also plotted is the best fit line for $2 < t < 10$. The expected slope as seen experimentally by Hnat and Buckmaster [18] is 1 and the computed slope here is 1.003. The maximum volume fluctuation of the gas bubble using the CLSVOF method is 0.0009%. The maximum volume fluctuation using the LS method with redistancing and comparable resolution is about 20% (see [29]).

6.7. Gas bubble bursting at a free surface. Here, we compute the problem of a gas bubble rising to the free surface of a liquid. When the bubble breaks through the surface, large surface tension forces are produced which ultimately cause a jet of liquid to be ejected. This jet of liquid can subsequently break up into drops.

This problem was studied by Boulton-Stone and Blake [9] using the boundary integral method and by Sussman and Smereka [32] using the LS method. In our CLSVOF computation, we consider a spherical 4-mm air bubble released just below the surface. The domain (axisymmetric geometry) is $\Omega = \{(x, y) \mid 0 \le r \le 3 \text{ and } 0 \le z \le 12\}$ and the mesh is 48×192. For these computations the density ratio is 1000 : 1 and the viscosity ratio is 100 : 1. The relevant (dimensionless) parameters for this problem are $Re = 474$, $\mu_l/\mu_g = 100$, $\rho_l/\rho_g = 1000$, $Fr = 0.64$, and $We = 1.0$. The interfacial thickness parameter α represents two grid cells. In Fig. 15 we show the evolution of the air bubble rising to the surface of the air/water interface and then bursting due to stiff surface tension effects. The jet then breaks up due to capillary instabilities, emitting satellite drops. The maximum volume fluctuation of the water for this problem is 0.007% even through the complex changes in topology.

7. THREE-DIMENSIONAL RESULTS

In Fig. 16, we show the computation of the rise of a fully three-dimensional inviscid air bubble in water. The density ratio is 816 : 1 and the dimensionless parameters used

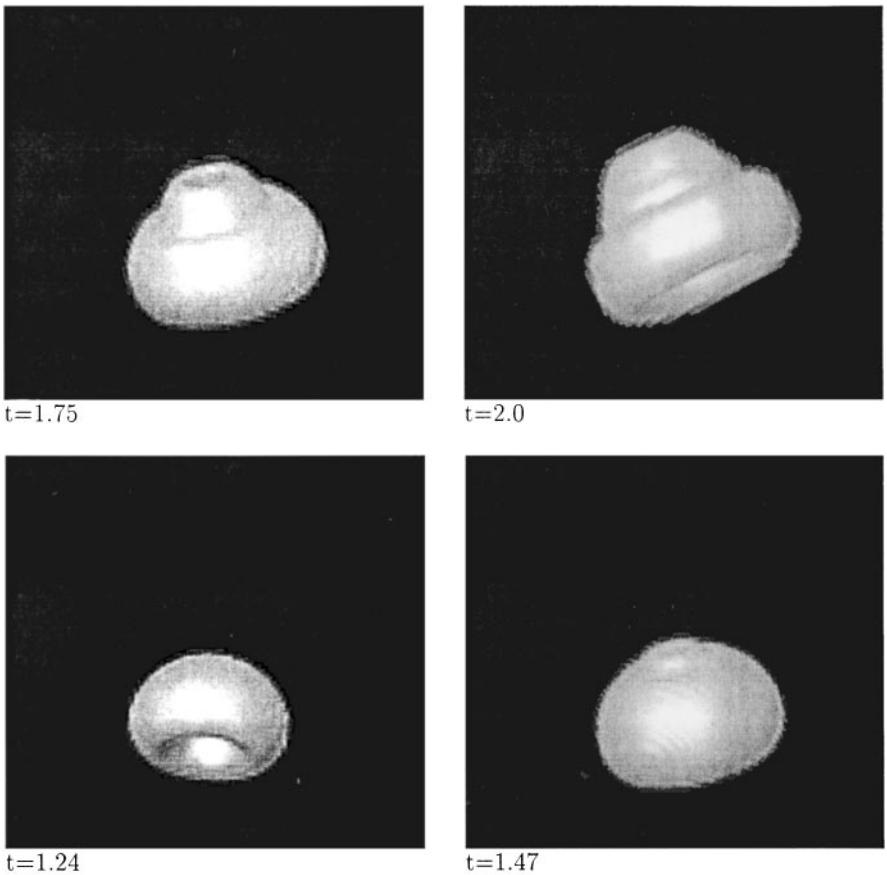

FIG. 16. Rise of inviscid air bubble in water. $We = 200$, effective fine grid $64 \times 64 \times 128$.

for this problem are $We = 200$ and $Fr = 1$. We use the adaptive mesh refinement algorithm described in [29] when computing this problem. The dimensions of the domain are $4 \times 4 \times 8$ and the mesh size on the finest level of adaptivity is $\Delta x = \Delta y = \Delta z = 1/16$. We use far-field boundary conditions on all sides of the domain. In Fig. 17, we display a cross-section of the bubble at $t = 1.24$ and $t = 1.48$ and compare these results with the results computed using the axisymmetric version of the CLSVOF algorithm with the same fine grid resolution $\Delta r = 1/16$. The maximum mass fluctuation of the gas bubble for this 3D computation is 0.01%.

In Fig. 18, we display the interaction of two viscous gas bubbles in liquid. For this problem, we start off with two gas bubbles whose centers are offset in the "x" direction by one bubble radii and offset in the "z" direction by 2.3 radii. The density ratio is 20 : 1 and the viscosity ratio is 26 : 1. The dimensionless parameters we use for this problem are $We = 50$, $Fr = 1$, and $Re = 50^{3/4}$. Except for the initial bubble offsets (not reported by [35]) these parameters correspond to Fig. 12 in [35]. We again use adaptive mesh refinement when computing this problem. The dimensions of the domain are $4 \times 4 \times 8$ and the mesh size on the finest level of adaptivity is $\Delta x = \Delta y = \Delta z = 1/16$. We use free-slip boundary conditions on all sides of the domain. Our results agree qualitatively with those in [35].

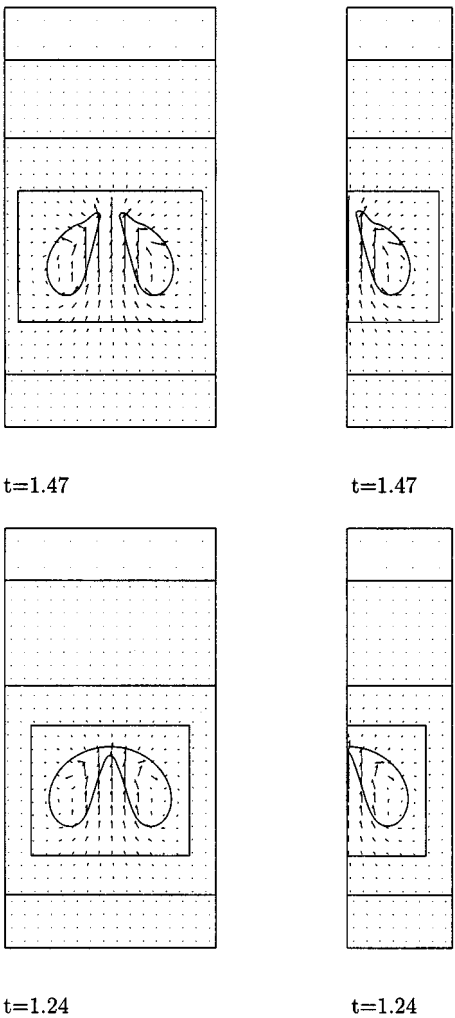

FIG. 17. Spherical gas bubble in liquid; density ratio 816 : 1; We = 200. Left, cross section of three-dimensional results (y = 2, x-z plane), effective fine grid $64 \times 64 \times 128$, dimensions of domain: $4 \times 4 \times 8$. Right, axisymmetric results, effective fine grid 32×128, dimensions of domain 2×8.

8. CONCLUSIONS

We have developed a coupled level set/volume-of-fluid (CLSVOF) method for representing the free-surface in two-phase flow problems. Our goal in designing this method was to produce an algorithm that is more accurate then either the level set or volume-of-fluid method alone, especially when computing problems for which surface tension and changes in the topology of the free surface are dominant features of the flow.

We have tested the CLSVOF method on two-dimensional, axisymmetric, and fully three-dimensional flows. We show that for all of these problems, the CLSVOF method produces results that are as good as or better than those produced by our comparison methods. For example, for the Rayleigh capillary problem (see Subsection 6.4) the CLSVOF method and LS method give identical results. On the other hand, for problems in which the interface develops corners, or there is interfacial merging and pinching, the CLSVOF method

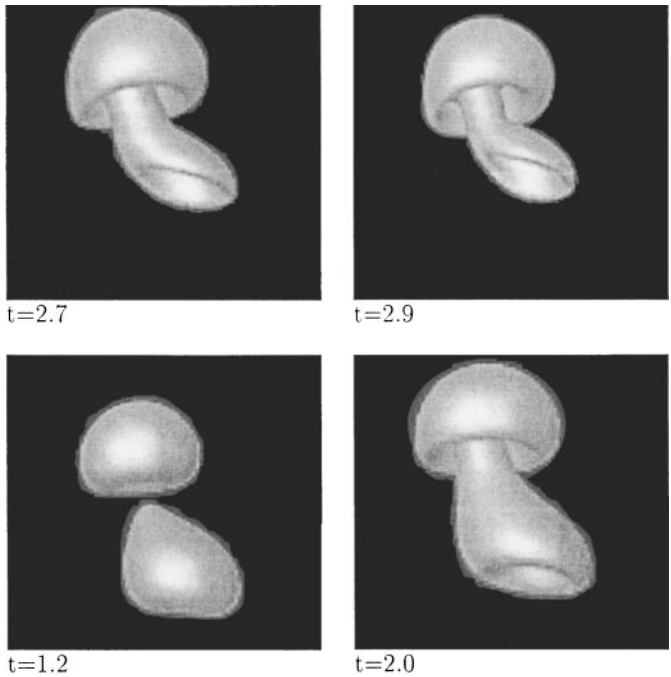

FIG. 18. Non-axisymmetric merging of two viscous gas bubbles, effective fine grid $64 \times 64 \times 128$.

outperforms the LS method, since the CLSVOF method conserves mass to a fraction of a percent while the LS method can lose as much as 20% of mass.

We also demonstrate that on problems such as translating and rotating circles and falling droplets, the CLSVOF method produces results which are as good as or better than those produced with the best VOF methods (e.g., methods which employ piecewise linear interface construction and second order advection algorithms.) Furthermore, for problems with surface tension the CLSVOF method is generally superior to these VOF methods, since the computation of the curvature of the interface is easier as it is simply the Laplacian of the level set function ϕ.

As mentioned in the Introduction, we were motivated to develop the CLSVOF method in order to improve our computations of micro-scale jetting devices. Based on the examples presented in Section 6, we believe that we have achieved our goal. For example, we have shown that the CLSVOF method conserves mass as well as or better than the LS method while retaining the advantage of the simpler—and probably more accurate—method of computing surface tension that characterize level set methods. In particular, we have used the CLSVOF method to compute the Rayleigh capillary instability with parameters typical of a micro-scale jetting problem and shown that the numerical solution converges to first-order, both before and after the cylindrical tube breaks up into drops (see Subsection 6.4).

9. APPENDIX

9.1. Approximation of the advection term. In this section, we describe the discretization of the advection term,

$$[(\boldsymbol{U} \cdot \nabla)\boldsymbol{U}]^{n+\frac{1}{2}}. \tag{71}$$

In the process of describing the discretization of (71) we shall also describe how the discretely divergence free advective edge velocities, $u^{ADV,n+1/2}_{i+1/2,\,j}$ and $v^{ADV,n+1/2}_{i,\,j+1/2}$, are formed.

The discretization of (71) in this algorithm is very similar to the discretization used by [3, 23]. It is a predictor-corrector method based on the unsplit Godunov method introduced by Colella [14].

In the predictor we extrapolate the velocity $\boldsymbol{U}$ to the cell faces at $t^{n+1/2}$ using a second-order Taylor series expansion in space and time. The time derivative $\boldsymbol{U}_t$ is replaced using (7). For face $(i+1/2,\ j)$ this gives

$$\boldsymbol{U}^{n+1/2,L}_{i+1/2,\,j} = \boldsymbol{U}^n_{ij} + \left(\frac{\Delta r}{2} - \frac{u^n_{ij}\Delta t}{2}\right)\boldsymbol{U}^n_{x,ij} - \frac{\Delta t}{2}(v\widehat{\boldsymbol{U}}_y)_{ij} - \frac{\Delta t}{2}(w\widehat{\boldsymbol{U}}_z)_{ij} + \frac{\Delta t}{2}\left(-\frac{Gp^{n-1/2}_{ij}}{\rho^n_{ij}} + \frac{\mathcal{L}^n_{ij}}{\rho^n_{ij}} - \frac{\mathcal{M}^n_{ij}}{\rho^n_{ij}} + \boldsymbol{F}\right) \tag{72}$$

extrapolated from cell $(i,\ j)$, and

$$\boldsymbol{U}^{n+1/2,R}_{i+1/2,\,j} = \boldsymbol{U}^n_{i+1,\,j} - \left(\frac{\Delta r}{2} + \frac{u^n_{i+1,\,j}\Delta t}{2}\right)\boldsymbol{U}^n_{x,i+1,\,j} - \frac{\Delta t}{2}(v\widehat{\boldsymbol{U}}_y)_{i+1,\,j} - \frac{\Delta t}{2}(w\widehat{\boldsymbol{U}}_z)_{i+1,\,j} + \frac{\Delta t}{2}\left(-\frac{Gp^{n-1/2}_{i+1,\,j}}{\rho^n_{i+1,\,j}} + \frac{\mathcal{L}^n_{i+1,\,j}}{\rho^n_{i+1,\,j}} - \frac{\mathcal{M}^n_{i+1,\,j}}{\rho^n_{i+1,\,j}} + \boldsymbol{F}\right) \tag{73}$$

extrapolated from cell $(i+1,\ j)$.

Analogous formulae are used to predict values at each of the other faces of the cell

$$\boldsymbol{U}^{n+1/2,T/B}_{i,\,j+1/2}. \tag{74}$$

The first derivative normal to the face, $\boldsymbol{U}^n_x$ for the example in (72) and (73), is evaluated using a monotonicity-limited fourth-order slope approximation [13]. The limiting is done on each component of the velocity at t^n individually.

The transverse derivative term,

$$v\widehat{\boldsymbol{U}}_y,$$

is evaluated by first extrapolating $\boldsymbol{U}$ to the transverse faces from the cell centers on either side, using normal derivatives only, and then choosing between these states using the upwinding procedure as described in detail by Almgren *et al.* [3] and Puckett *et al.* [23].

Once we have computed $u^{n+1/2,L/R}_{i+1/2,\,j}$ and $v^{n+1/2,T/B}_{i,\,j+1/2}$, we are in a position to construct the normal face-centered edge velocities at $t^{n+1/2}$,

$$u^{ADV}_{i+1/2,\,j},\ v^{ADV}_{i,\,j+1/2}.$$

Given $u^{n+1/2,L}_{i+1/2,\,j}$ and $u^{n+1/2,R}_{i+1/2,\,j}$, we use an upwinding procedure to choose $u^{n+1/2}_{i+1/2,\,j}$,

$$u^{n+1/2}_{i+1/2,\,j} = \begin{cases} u^L & \text{if } u^L > 0 \quad \text{and} \quad u^L + u^R > 0 \\ 0 & \text{if } u^L \le 0, u^R \ge 0 \quad \text{or} \quad u^L + u^R = 0 \\ u^R & \text{if } u^R < 0 \quad \text{and} \quad u^L + u^R < 0. \end{cases} \tag{75}$$

Here, we suppress the $i+1/2,\ j$ spatial indices on left and right states and we also suppress the $n+1/2$ temporal index.

We follow a similar procedure as in (75) to construct $v^{n+1/2}_{i,\,j+1/2}$.

These normal velocities on cell faces at $t^{n+1/2}$,

$$u^{n+1/2}_{i+1/2,\,j},\ v^{n+1/2}_{i,\,j+1/2}, \tag{76}$$

are second-order accurate but do not, in general, satisfy the discrete divergence-free condition. In order to make these velocities divergence-free, we apply the MAC projection [6]. The equation

$$D^{MAC}\left(\frac{1}{\rho^n}G^{MAC}p^{MAC}\right) = D^{MAC}\left(\boldsymbol{U}^{n+\frac{1}{2}}\right) \tag{77}$$

is solved for p^{MAC}, where

$$D^{MAC}\boldsymbol{U}^{n+\frac{1}{2}} = \frac{r_{i+1/2}u^{n+\frac{1}{2}}_{i+1/2,\,j} - r_{i-1/2}u^{n+\frac{1}{2}}_{i-1/2,\,j}}{r_i\,\Delta r} + \frac{v^{n+\frac{1}{2}}_{i,\,j+1/2} - v^{n+\frac{1}{2}}_{i,\,j-1/2}}{\Delta z}$$

and $G^{MAC} = -(D^{MAC})^T$ so that

$$\left(G^{MAC}_x p^{MAC}\right)_{i+\frac{1}{2},\,j} = \frac{\left(p^{MAC}_{i+1,\,j} - p^{MAC}_{i,\,j}\right)}{\Delta r}$$

with G^{MAC}_y defined analogously. The resulting linear system (77) is solved using a multigrid preconditioned conjugate gradient solver [34].

The face-based advection velocities at $t^{n+1/2}$ are then defined by

$$u^{ADV}_{i+1/2,\,j} = u^{n+\frac{1}{2}}_{i+1/2,\,j} - \frac{1}{\rho^n_{i+1/2,\,j}}\left(G^{MAC}_x p^{MAC}\right)_{i+\frac{1}{2},\,j} \tag{78}$$

with $v^{ADV}_{i,\,j+1/2}$ defined analogously. The quantity $\rho^n_{i+1/2,\,j}$ in (78) is defined by

$$\rho^n_{i+1/2,\,j} = \frac{1}{2}\left(\rho^n_{ij} + \rho^n_{i+1,\,j}\right)$$

with $\rho^n_{i,\,j+1/2}$ defined analogously.

The next step, after constructing the advective velocities

$$u^{ADV}_{i+1/2,\,j},\ v^{ADV}_{i,\,j+\frac{1}{2}},$$

is to choose the appropriate state $\boldsymbol{U}^{n+1/2}_{i+1/2,\,j}$ given the left and right states in (72) through (73),

$$\boldsymbol{U}^{n+1/2,\,L}_{i+1/2,\,j},\ \boldsymbol{U}^{n+1/2,\,R}_{i+1/2,\,j}.$$

We have

$$\boldsymbol{U}^{n+1/2}_{i+1/2,\,j} = \begin{cases} \boldsymbol{U}^L & \text{if } u^{ADV} > 0 \\ \frac{1}{2}(\boldsymbol{U}^L + \boldsymbol{U}^R) & \text{if } u^{ADV} = 0 \\ \boldsymbol{U}^R & \text{if } u^{ADV} < 0. \end{cases} \tag{79}$$

Here, we suppress the $i+1/2,\ j$ spatial indices on left and right states and we also suppress the $n+1/2$ temporal index.

We follow a similar procedure as in (79) to construct

$$\boldsymbol{U}_{i,\,j+\frac{1}{2}}^{n+\frac{1}{2}}.$$

The advection term can now be defined by

$$[(\boldsymbol{U}\cdot\nabla)\boldsymbol{U}]_{i,\,j}^{n+\frac{1}{2}} = \frac{1}{r_i\,\Delta r}\,\frac{r_{i+1/2}u_{i+1/2,\,j}^{ADV} + r_{i-1/2}u_{i-1/2,\,j}^{ADV}}{2}\left(\boldsymbol{U}_{i+1/2,\,j} - \boldsymbol{U}_{i-1/2,\,j}\right) + \frac{1}{\Delta z}\,\frac{v_{i,\,j+1/2}^{ADV} + v_{i,\,j-1/2}^{ADV}}{2}\left(\boldsymbol{U}_{i,\,j+1/2} - \boldsymbol{U}_{i,\,j-1/2}\right). \tag{80}$$

9.2. Discretization of pressure gradient, viscous, and surface tension terms. In this section we describe the finite difference approximation to the pressure gradient, Gp, viscous term, $\mathcal{L}$, and surface tension term, $\mathcal{M}$.

The discrete pressure gradient is defined by

$$(Gp)_{ij} \equiv \begin{pmatrix} \dfrac{p_{i+\frac{1}{2},\,j+\frac{1}{2}} + p_{i+\frac{1}{2},\,j-\frac{1}{2}} - p_{i-\frac{1}{2},\,j+\frac{1}{2}} - p_{i-\frac{1}{2},\,j-\frac{1}{2}}}{2\Delta r} \\ \dfrac{p_{i+\frac{1}{2},\,j+\frac{1}{2}} - p_{i+\frac{1}{2},\,j-\frac{1}{2}} + p_{i-\frac{1}{2},\,j+\frac{1}{2}} - p_{i-\frac{1}{2},\,j-\frac{1}{2}}}{2\Delta z} \end{pmatrix}, \tag{81}$$

where G here denotes a discrete gradient operator defined at cell centers but operating on nodal data.

The first component of the viscous term $\nabla\cdot 2\mu(\phi)\mathcal{D}$ is discretized as

$$(\mathcal{L})_{ij}^1 = \begin{pmatrix} \dfrac{2\mu_{i+\frac{1}{2},\,j}r_{i+1/2}(u_{i+1,\,j}-u_{i,\,j}) - 2\mu_{i-1/2,\,j}r_{i-1/2}(u_{i,\,j}-u_{i-1,\,j})}{r_i\,\Delta r^2} \\ + \dfrac{\mu_{i,\,j+1/2}(u_{i,\,j+1}-u_{i,\,j}) - \mu_{i,\,j-1/2}(u_{i,\,j}-u_{i,\,j-1})}{\Delta z^2} \\ + \dfrac{\mu_{i,\,j+1/2}(v_{i+1,\,j+1}-v_{i-1,\,j+1}+v_{i+1,\,j}-v_{i-1,\,j}) - \mu_{i,\,j-1/2}(v_{i+1,\,j}-v_{i-1,\,j}+v_{i+1,\,j-1}-v_{i-1,\,j-1})}{\Delta r\,\Delta z} - \mu_{i,\,j}u_{i,\,j}/r_i^2 \end{pmatrix},$$

where

$$\mu_{i+\frac{1}{2},\,j} = \frac{1}{2}(\mu(\phi_{i,\,j}) + \mu(\phi_{i+1,\,j})) \qquad \text{and} \qquad \mu_{i,\,j+\frac{1}{2}} = \frac{1}{2}(\mu(\phi_{i,\,j}) + \mu(\phi_{i,\,j+1})). \tag{82}$$

Remarks.

- The discretization described by (82) is a basic discretization; improved discretizations for variable viscosity problems are described by Coward *et al.* [15].
- The second component of the viscous term, $(\mathcal{L})_{ij}^2$, is discretized in a similar manner as the first component.

The surface tension term $\gamma\kappa(\phi)\nabla H(\phi)$ is discretized as

$$(\mathcal{M})_{ij} \equiv \gamma(D\boldsymbol{N})_{ij}(GH^{node})_{ij}. \tag{83}$$

$\boldsymbol{N}_{i+1/2,\,j+1/2}$ is the discrete approximation of the level set normal $\nabla\phi/|\nabla\phi|$,

$$\boldsymbol{N}_{i+1/2,\,j+1/2} \equiv \frac{(G\phi)_{i+1/2,\,j+1/2}}{\left|(G\phi)_{i+1/2,\,j+1/2}\right|}, \tag{84}$$

where

$$(G\phi)_{i+1/2,\,j+1/2} \equiv \begin{pmatrix} \frac{\phi_{i+1,j+1}+\phi_{i+1,j}-\phi_{i,j+1}-\phi_{i,j}}{2\Delta r} \\ \frac{\phi_{i+1,j+1}-\phi_{i+1,j}+\phi_{i,j+1}-\phi_{i,j}}{2\Delta z} \end{pmatrix}. \tag{85}$$

Here we use G to refer to the discrete gradient operator defined on nodes but operating on cell-centered data.

We define the cell-based discrete divergence operator D by

$$(D\boldsymbol{N})_{ij} \equiv \frac{r_{i+1/2}\left(n^1_{i+1/2,\,j-1/2}+n^1_{i+1/2,\,j+1/2}\right)-r_{i-1/2}\left(n^1_{i-1/2,\,j-1/2}+n^1_{i-1/2,\,j+1/2}\right)}{r_i\,\Delta r} + \frac{r_{i+1/2}\left(n^2_{i+1/2,\,j+1/2}-n^2_{i+1/2,\,j-1/2}\right)+r_{i-1/2}\left(n^2_{i-1/2,\,j+1/2}-n^2_{i-1/2,\,j-1/2}\right)}{r_i\,\Delta z}. \tag{86}$$

The node-based Heaviside function $H^{node}_{i+1/2,\,j+1/2}$ is defined as

$$H^{node}_{i+\frac{1}{2},\,j+\frac{1}{2}} = H\left(\frac{\phi_{i+1,j}+\phi_{i,j}+\phi_{i+1,j+1}+\phi_{i,j+1}}{4}\right). \tag{87}$$

9.3. Discretization of the projection. In this section we describe the discrete "approximate projection," $\mathcal{P}$, which is used in (59). $\mathcal{P}$ is an approximation to the projection operator $\boldsymbol{P}$ described in (21). We remark that a detailed description of the approximate projection is given by [4].

Given the discrete vector field

$$\frac{\boldsymbol{U}^*-\boldsymbol{U}^n}{\Delta t}, \tag{88}$$

we decompose (88) into an *approximately* divergence free part

$$\frac{\boldsymbol{U}^{n+1}-\boldsymbol{U}^n}{\Delta t} \tag{89}$$

and the discrete gradient of a scalar q divided by density

$$\frac{(Gq)_{ij}}{\rho^{n+1/2}_{ij}}, \tag{90}$$

where the discrete gradient G in (90) is defined in (81).

The approximate projection is computed by solving

$$L_\rho q = D\left(\frac{\boldsymbol{U}^*-\boldsymbol{U}^n}{\Delta t}\right) \tag{91}$$

for q. The right hand side of (91) is an approximation to $\nabla\cdot\boldsymbol{V}$ found in the right hand side of (19). The discrete divergence $D\boldsymbol{U}$ is

$$(D\boldsymbol{U})_{i+\frac{1}{2},\,j+\frac{1}{2}} = \frac{r_{i+1}(u_{i+1,j}+u_{i+1,j+1})-r_i(u_{i,j}+u_{i,j+1})}{r_{i+1/2}\Delta r} + \frac{r_{i+1}(v_{i+1,j+1}-v_{i+1,j})-r_i(v_{i,j}-v_{i,j+1})}{r_{i+1/2}\Delta z}. \tag{92}$$

The left hand side of (91), $L_\rho q$, is an approximation to $\nabla \cdot \frac{1}{\rho}\nabla p$ found in the left hand side of (19). The discrete representation of $L_\rho q$ is

$$(L_\rho q)_{i+\frac{1}{2},j+\frac{1}{2}} = \frac{1}{6r_{i+1/2}h^2}\begin{pmatrix} \frac{r_i}{\rho_{i,j}}\left(2q_{i-\frac{1}{2},j-\frac{1}{2}} + q_{i+\frac{1}{2},j-\frac{1}{2}} + q_{i-\frac{1}{2},j+\frac{1}{2}} - 4q_{i+\frac{1}{2},j+\frac{1}{2}}\right) \\ + \frac{r_i}{\rho_{i,j+1}}\left(2q_{i-\frac{1}{2},j+3/2} + q_{i+\frac{1}{2},j+3/2} + q_{i-\frac{1}{2},j+\frac{1}{2}} - 4q_{i+\frac{1}{2},j+\frac{1}{2}}\right) \\ + \frac{r_{i+1}}{\rho_{i+1,j}}\left(2q_{i+3/2,j+\frac{1}{2}} + q_{i+\frac{1}{2},j-\frac{1}{2}} + q_{i+3/2,j+\frac{1}{2}} - 4q_{i+\frac{1}{2},j+\frac{1}{2}}\right) \\ + \frac{r_{i+1}}{\rho_{i+1,j+1}}\left(2q_{i+3/2,j+3/2} + q_{i+\frac{1}{2},j+3/2} + q_{i+3/2,j+\frac{1}{2}} - 4q_{i+\frac{1}{2},j+\frac{1}{2}}\right) \end{pmatrix}. \tag{93}$$

The operator $L_\rho q$ (93) is derived from the variational form of (19),

$$\int \frac{1}{\rho}\nabla q(\mathbf{x}) \cdot \nabla \psi(\mathbf{x})\, d\mathbf{x} = \int \frac{\boldsymbol{U}^* - \boldsymbol{U}^n}{\Delta t} \cdot \nabla \psi(\mathbf{x})\, d\mathbf{x}, \qquad \forall \psi(\mathbf{x}), \tag{94}$$

where $d\mathbf{x}$ is the volume element $r\, dr\, d\theta$. The finite element basis functions $\psi(\mathbf{x})$ represent standard piecewise bilinear functions.

After (91) is solved, we form $(\boldsymbol{U}^{n+1} - \boldsymbol{U}^n)/\Delta t$,

$$\frac{\boldsymbol{U}^{n+1} - \boldsymbol{U}^n}{\Delta t} = \frac{\boldsymbol{U}^* - \boldsymbol{U}^n}{\Delta t} - \frac{Gq}{\rho^{n+1/2}}, \tag{95}$$

and $p^{n+1/2}$,

$$p^{n+1/2} = p^{n-1/2} + q.$$

Remarks.

- The discrete projection step presented here is slightly different from the continuous analogue presented in Subsection 2.1 because we are solving for the *difference* in pressure $q = p^{n+1/2} - p^{n-1/2}$ instead of the actual pressure $p^{n+1/2}$.
- The discrete projection operator $\mathcal{P}$ is called an approximate projection because the discrete divergence of (89),

$$\left[D\left(\frac{\boldsymbol{U}^{n+1} - \boldsymbol{U}^n}{\Delta t}\right)\right]_{i+1/2,j+1/2}, \tag{96}$$

is not identically zero. In order to see why (96) is not necessarily zero, we apply the discrete divergence D to both sides of (95) in order to arrive at

$$\left[D\left(\frac{\boldsymbol{U}^{n+1} - \boldsymbol{U}^n}{\Delta t}\right)\right]_{i+1/2,j+1/2} = \left[D\left(\frac{\boldsymbol{U}^* - \boldsymbol{U}^n}{\Delta t}\right)\right]_{i+1/2,j+1/2} - \left[D\frac{1}{\rho^{n+1/2}}Gq\right]_{i+1/2,j+1/2}. \tag{97}$$

The discrete operator $D(1/\rho^{n+1/2})Gq$ is not the same as $L_\rho q$ which means (96) is not necessarily zero.

REFERENCES

1. I. Aleinov and E. G. Puckett, Computing surface tension with high-order kernels, in *Proceedings of the 6th International Symposium on Computational Fluid Dynamics, Lake Tahoe, CA, 1995*.
2. R. R. Allen, J. D. Meyer, and W. R. Knight, Thermodynamics and hydrodynamics of thermal ink jets, *Hewlett-Packard J.* May (1985).
3. A. S. Almgren, J. B. Bell, P. Colella, L. H. Howell, and M. Welcome, A conservative adaptive projection method for the incompressible Navier–Stokes equations in three dimensions, *J. Comput. Phys.* **142**, 1–146 (1998).
4. A. S. Almgren, J. B. Bell, and W. G. Szymczak, A numerical method for the incompressible Navier–Stokes equations based on an approximate projection, *SIAM J. Sci. Comput.* **17**(2), (1996).
5. J. B. Bell, P. Colella, and H. M. Glaz, A second-order projection method for the incompressible Navier–Stokes equations, *J. Comput. Phys.* **85**, 257 (1989).
6. J. B. Bell, P. Colella, and L. H. Howell, An efficient second-order projection method for viscous incompressible flow, in *10th AIAA Computational Fluid Dynamics Conference, Honolulu, June 24–27, 1991*.
7. J. B. Bell and D. L. Marcus, A second-order projection method for variable-density flows, *J. Comput. Phys.* **101**, 334 (1992).
8. J. P. Best, The formation of toroidal bubbles upon the collapse of transient cavities, *J. Fluid Mech.* **251**, 79 (1993).
9. J. M. Boulton-Stone and Blake, Gas bubbles bursting at a free surface, *J. Fluid Mech.* **254**, 437 (1993).
10. A. Bourlioux, A coupled level-set volume-of-fluid algorithm for tracking material interfaces, in *Proceedings of the 6th International Symposium on Computational Fluid Dynamics, Lake Tahoe, CA, 1995*.
11. J. U. Brackbill, D. B. Kothe, and C. Zemach, A continuum method for modeling surface tension, *J. Comput. Phys.* **100**, 335 (1992).
12. Y. C. Chang, T. Y. Hou, B. Merriman, and S. Osher, Eulerian capturing methods based on a level set formulation for incompressible fluid interfaces, *J. Comput. Phys.* **124**, 449 (1996).
13. P. Colella, A direct Eulerian MUSCL scheme for gas dynamics, *SIAM J. Comput.* **6**, 104 (1985).
14. P. Colella, A multidimensional second order Godunov scheme for conservation laws, *J. Comput. Phys.* **87**, 171 (1990).
15. A. V. Coward, Y. Y. Renardy, M. Renardy, and J. R. Richards, Temporal evolution of periodic disturbances in two-layer couette flow, *J. Comput. Phys.* **132**, 346 (1997).
16. N. V. Deshpande, Fluid mechanics of bubble growth and collapse in a thermal ink-jet printer, in *SPSE/SPIES Electronic Imaging Devices and Systems Symposium, January 1989*.
17. H. Haj-Hariri, Q. Shi, and A. Borhan, Thermocapillary motion of deformable drops at finite reynolds and marangoni numbers, *Phys. Fluids* **9**, 845 (1997).
18. J. G. Hnat and J. D. Buckmaster, Spherical cap bubbles and skirt formation, *Phys. Fluids* **19**(2), 182 (1976).
19. H. Lamb, *Hydrodynamics* (Dover, New York, 1932).
20. T. S. Lundgren and N. N. Mansour, Vortex ring bubbles, *J. Fluid Mech.* **224**, 177 (1991).
21. S. Osher and J. A. Sethian, Fronts propagating with curvature-dependent speed: Algorithms based on hamilton-jacobi formulations, *J. Comput. Phys.* **79**(1), 12 (1988).
22. J. E. Pilliod and E. G. Puckett, Second order volume-of-fluid interface tracking algorithms, *J. Comput. Phys.*, in press.
23. E. G. Puckett, A. S. Almgren, J. B. Bell, D. L. Marcus, and W. G. Rider, A high-order projection method for tracking fluid interfaces in variable density incompressible flows, *J. Comput. Phys.* **130**, 269 (1997).
24. W. J. Rider and D. B. Kothe, *Stretching and Tearing Interface Tracking Methods*, AIAA paper 95-1717, 1995.
25. W. J. Rider, D. B. Kothe, S. Jay Mosso, J. H. Cerutti, and J. I. Hochstein, Accurate solution algorithms for incompressible multiphase flows, AIAA paper 95-0699, October 30, 1994.
26. G. Ryskin and L. G. Leal, Numerical solution of free boundary problems in fluid mechanics. Part 1. The finite-difference technique, *J. Fluid Mech.* **148**, 1 (1984).
27. G. Ryskin and L. G. Leal, Numerical solution of free boundary problems in fluid mechanics. Part 2. Buoyancy-driven motion of a gas bubble through a quiescent liquid, *J. Fluid Mech.* **148**, 19 (1984).

28. G. Strang, On the construction and comparison of difference schemes, *SIAM J. Numer. Anal.* **5**, 506 (1968).
29. M. Sussman, A. Almgren, J. Bell, P. Colella, L. Howell, and M. Welcome, An adaptive level set approach for incompressible two-phase flows, *J. Comput. Phys.* **148**, 81 (1999).
30. M. Sussman and E. Fatemi, An efficient, interface preserving level set redistancing algorithm and its application to interfacial incompressible fluid flow, *SIAM J. Sci. Comput.* **20**, 1165–1191 (1999).
31. M. Sussman, E. Fatemi, P. Smereka, and S. J. Osher, An improved level set method for incompressible two-phase flows, *J. Comput. Fluids* **27** (5–6), 663 (1998).
32. M. Sussman and P. Smereka, Axisymmetric free boundary problems, *J. Fluid Mech.* **341**, 269 (1997).
33. M. Sussman, P. Smereka, and S. J. Osher, A level set approach for computing solutions to incompressible two-phase flow, *J. Comput. Phys.* **114**, 146 (1994).
34. O. Tatebe, The multigrid preconditioned conjugate gradient method, in *6th Copper Mountain Conference on Multigrid Methods, Copper Mountain, CO, April 4–9, 1993*.
35. S. O. Unverdi and G. Tryggvason, A front-tracking method for viscous, incompressible, multi-fluid flows, *J. Comput. Phys.* **100**, 25 (1992).
36. S. T. Zalesak, Fully multidimensional flux-corrected transport algorithms for fluids, *J. Comput. Phys.* **31**, 335 (1979).